

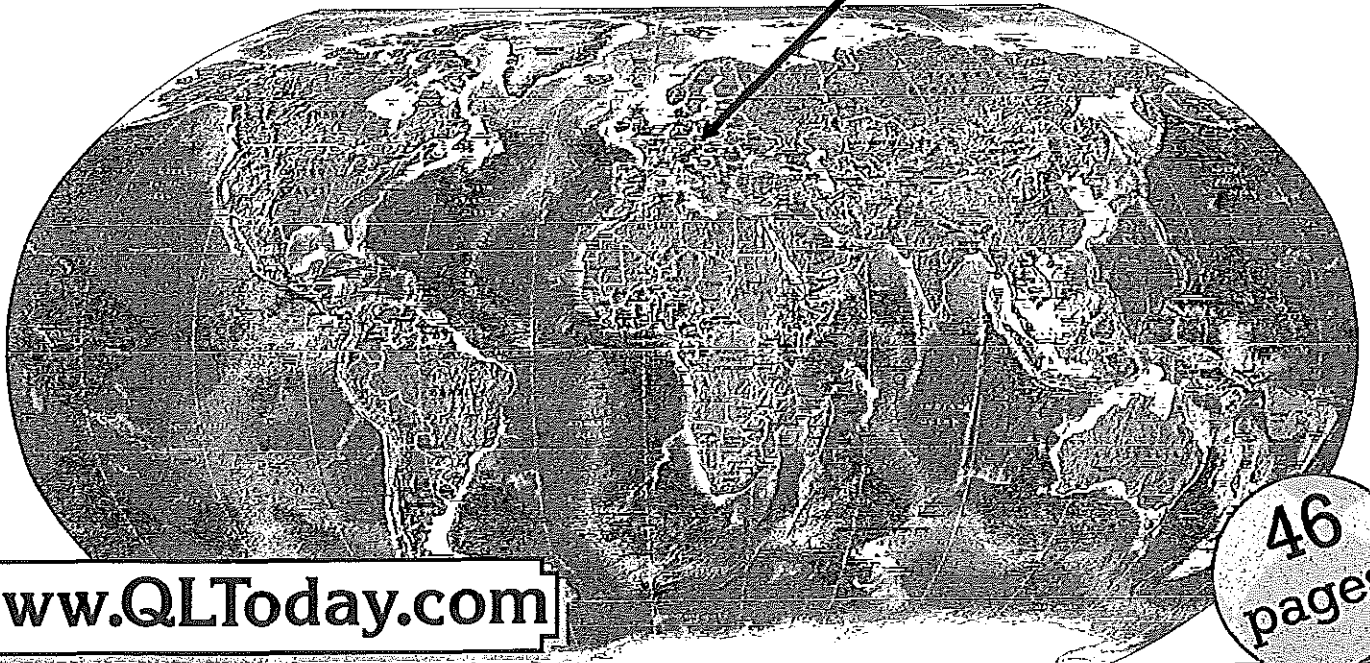
QL Today

Volume 15
Issue 1
Sept. - Nov.
2010

ISSN 1432-5454

The Magazine about QL, QDOS,
Sinclair Computers, SMSQ...

VIENNA



www.QLToday.com

46
pages!

Now was Vienna in 2010?

Great!

More inside this issue...

And the next show? Where?
When? Your feedback please!

Contents

- 3 Editorial
- 4 News
- 6 Nodding Donkeys, Steam, Air, Wind and Oil
Tony Firshman
- 10 Goodbye Paper? Geoff Wicks
- 11 25 Years - Part 4 Tony Tebby
- 17 Zoostorm Freedom XL 10-270
Dilwyn Jones
- 19 The Future of the QL Geoff Wicks
- 28 Easy PEasy - Part 2 Norman Dunbar
- 40 Artificial Intelligence Stephen Poole
- 45 QL Today Volume 15 & Meetings
Jochen Merz

Advertisers

in alphabetical order

Jochen Merz Software	Address sheet
Quanta	39
QuoVadis Design	21

**The deadline for
the next issue
is the 15th of
November 2010**

**We look forward
to receive and
print YOUR article
in time!**

QL Today

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software Tel. +49 203 502011
Kaiser-Wilhelm-Str. 302 Fax +49 203 502012
47169 Duisburg email: smsq@j-m-s.com
Germany email: QLToday@j-m-s.com

Editor:

Geoff Wicks Tel. +44 1332 271366
Flat 5b email: gtwicks@btinternet.com
Wordsworth Avenue email: QLToday@j-m-s.com
Derby DE24 9HQ
United Kingdom

Co-Editor & UK Office:

Bruce Nicholls Tel +44 20 71930539
38 Derham Gardens Fax +44 870 0568755
Upminster email: qltoday@q-v-d.demon.co.uk
Essex RM14 3HA email: QLToday@j-m-s.com
United Kingdom

QL Today is published four times a year, our volume begins on beginning of June. Please contact the German or English office for current subscription rates or visit our homepage www.QLTODAY.com.

We welcome your comments, suggestions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email. We prefer ASCII, Quill or text87 format. Word .DOC is also fine, and so is PDF. If you send Word .DOC or PDF, then please send pictures separately as well. Pictures may be in .SCR format, we can also handle GIF, TIF, PNG or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

QL Today reserves the right to publish or not publish any material submitted. Under no circumstances will **QL Today** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL Today**. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

This magazine and all material within is Copyright 2010 Jochen Merz Software unless otherwise stated. Written permission is required from the publisher before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

If you need more information about the UNZIP program which is used by our BOOT program to unpack the files, we suggest that you visit Dilwyn Jones' web site where you find more information about lots of interesting QDOS software and INFOZIP at <http://www.dilwyn.uk6.net/arch/index.html>

Recently my editorials have been backfiring.

Last year I wrote the party was over for QL shows and, practically before the ink was dry, came news of a show in Austria - the third continental show in three years. By all accounts it was a great success. This year I have twice written that the QL-users email group was turning into a technical helpline, but two months ago it burst into life with a discussion on the future of the QL.

Subscribers to the QL-users group will have a sense of déjà vu when they read this issue because we have devoted several pages to a detailed report of the discussion. I make no apology for this because, as I have explained before, not all of our readers subscribe to the QL-users group. The discussion was too important for them to miss. Subscribers to the group will find there is much to gain from a second reading of the discussion. Three weeks of opinions have been compressed into 6,000 words giving a more permanent record than is possible in cyberspace.

The danger of internet forums is that they are ephemeral. A discussion takes place today and is forgotten tomorrow. Opinions expressed often do not lead to action. The discussion on the future of the QL must not be brushed aside in this way. It brought out the best in the QL community with, at times, challenging and hard hitting comments made in a spirit that was neither negative nor rancorous. We hope our readers will continue the discussion through the columns of QL Today.

Whatever the future of the QL it is looking increasingly likely that it will be a future without Quanta. The organisation has been telling us for months that under its constitution the treasurer had to leave the committee for at least a year, but then reappointed him treasurer within an hour of his stepping down at the AGM. Common sense would suggest that something doubtful has taken place. Neither the members nor the committee emerge with any credit from this story. The members have abdicated responsibility for Quanta, and the committee are unwilling - or, more seriously, unable - to justify their actions under the constitution. And to add to Quanta's woes the organisation is again without an editor for its magazine. Both matters have brought the demise of Quanta much nearer.

Some Quanta members will be angry on reading the last paragraph, but indignation does not solve problems. Action does. Quanta is still a large organisation. 177 members twenty six years into the life of a "failed" computer is a remarkable achievement of which Quanta can be justly proud. Although many of the members will now be too elderly to take an active role in the organisation, there is still enough younger blood for Quanta to be a vibrant organisation should the members so wish. The future of Quanta lies not in a QL Today editorial. Nor any longer in the hands of the committee. It lies firmly in the hands of its members. They must soon decide if Quanta is to live or Quanta is to die.

Back to Square One

John Gilpin had scarcely stepped down from the Quanta committee before finding himself back in the driving seat as treasurer of the organisation. The new committee co-opted him as treasurer in spite of previous warnings he was no longer eligible for the post. John Gilpin's re-appointment appears to be in breach of the Quanta constitution and, on the deadline date for this issue of QL Today, the Quanta website was still saying he could no longer remain as treasurer. The committee meeting at which the decision was made started with just two members present - Quanta's lowest permitted committee quorum.

It is Quanta's usual practice to hold a committee meeting immediately following its AGM, but this year only three members of the old committee were present at the AGM, one of whom was John Gilpin, who stepped down. One of the first actions of the new committee was to co-opt 3 additional members Tony Hill, Dave Buckley and John Gilpin.

Earlier in the AGM Chairwoman Sarah Gilpin had simply announced that Quanta had no treasurer and gave no indication of how the committee proposed to solve the problem. Later members were informed in the Quanta Magazine:

"As there had been no nomination for Treasurer/ Membership Secretary John Gilpin has been co-opted for one year only to continue in these posts".

In 2005 amendments to the Quanta constitution imposed a limit of six years during which members could serve on the committee. Events this year have exposed a weakness in these amendments in that they are either too complicated to understand or not practical for Quanta's present situation.

A year ago Geoff Wicks, writing as editor of QL Today, expressed the opinion in the magazine that the Quanta committee had misunderstood the constitution and that John Gilpin did not have to stand down until 2012. Earlier this year, this time writing as an individual member of Quanta, he posted on the QL-users email group the detailed legal arguments that led him to this conclusion. Neither Quanta nor anyone else responded with an alternative interpretation.

Quanta currently maintains that the six year rule applies equally to all members of the committee, but Geoff Wicks argues that the constitution makes a clear distinction between ordinary com-

mittee members and officers. The latter are permitted to serve out their full 3 year term of office before being required to step down. John Gilpin first joined the Quanta committee in 2002 and thus his six years ended in 2008.

John Gilpin is also back in his role as acting editor of the Quanta Magazine after the newly appointed editor stepped down after producing just one issue and also resigned from the committee. Quanta has not been able to find a permanent editor of its magazine since 2005.

Quanta is also reporting a lack of interest in two of its surveys. Only 23 out of 177 Quanta members responded to a last year's survey on a suggested merger between Quanta and QL Today. The proposal was overwhelmingly rejected. The more recent general survey has had a response from only about 50 UK QL-ers. A similar survey 6 years ago had a total of 124 respondents with 72 being Quanta members and 52 non-Quanta members. On that occasion to encourage a good response Quanta had entered all respondents in a draw with 3 prizes available to Quanta members and 3 to non-Quanta members.

Pictures Galore

In what is believed to be a first for a QL show Anton Preinsack has posted a long video report of the Austrian show on UTube:

<http://www.youtube.com/watch?v=O4Q7IUMkVRM>



The video is just over 6 minutes long and shows some of the hardware displayed at the show, a short scene showing the menus of QemuLator for MacOSX and some of the people who attended the show.

Not to be outdone Tony Firshman has posted 300 and Urs König 58 pictures of the show on their websites. Show organiser, Gerhard Plavec, has placed links to these on his site:

<http://kuel.org>

Website Makeover

Dilwyn Jones has done a complete makeover of his web site to give it a more modern and fresh look without affecting the content:

<http://www.dilwyn.me.uk/index.html>

OR

<http://dilwyn.me.uk/index.html>

As part of the makeover Dilwyn has placed a dedicated search engine on the site. In his own words:

"It's set up to concentrate on searching for Sinclair QL related pages, although it's still being fine-tuned and tweaked for best results.

It appears below the heading on my home page and when you enter a search term and click on Search, it expands the page with a list of results, like the usual search engine results. Those can be cleared when finished with by clicking on the X next to the SEARCH button.

If your browser's security settings restrict the page from running scripts and ActiveX controls (e.g. the beige bar which appears above the page in Internet Explorer) you should click where it tells you to, and tell it to let the code run.

This might be the case where the search box just says 'loading' and nothing seems to happen. I'd be interested in receiving feedback on whether you find it useful or not and any suggestions you may have!"

QL Today gave the search engine a field test by typing in the words "Just Words". A simple Google search gave no relevant hits in the first 20 results. Dilwyn's search engine gave as its first result the Just Words' website with four other "hits" on its first page.

In his contributions to the recent QL-users email group discussion on The Future of the QL Dilwyn mentioned that his advice pages get respectable number of hits and this is echoed by Just Words! Geoff Wicks comments:

"My help and advice page is the most visited section of the Just Words! website and this is something that has developed in the last twelve months. Clearly there is a need for QL help and advice on the web. The help and advice page is short and ideally I would like to expand it and add an interactive news section as well as rewrite the entire site and move to a new host. However time is the problem."



[Display with frames](#) | [Display without frames](#)

Top of Content Start

Search X

[Software Downloads](#)
[Documentation](#)
[About the QL](#)
[General information](#)
[e-Zines](#)
[Links to QL sites](#)
[Contact me](#)
[About me](#)

Welcome to Dilwyn Jones's QL information and PD software download pages. All of the material here is for the Sinclair QL and compatible computers only, unless otherwise stated. The site is split into sections - just click on the links in the box to the left to get to where you want.

The Sinclair QL

The Sinclair QL is a vintage computer first produced in 1983/4 and still used by many world-wide. If you are into retro-computing at all, the QL is an ideal choice of computer. It has a wonderful multi-tasking operating system and a great on-board SuperBASIC interpreter for those who like to try their hand at the odd bit of programming! The QL has spawned several compatible computers, including the Q40, Q60, Thor and Aurora, plus several QL emulators which run on computers as diverse as PCs, Amigas, Ataris, Macs and Linux systems. More information on the QL is available from the [About The QL](#) pages on this site.



PCB Updates

Dilwyn Jones has released details of the latest two updates of Malcolm Lear's PCB design program:

v7.15 27-05-10

- Logo improved and repositioned slightly in the vertical.
- Metric to imperial conversion calculator.
- Project loading selection no longer wraps.
- Art file format Version 0 (Pluto CAD) conversion improved.
- Area mirror and rotate no longer effect the zero offset. This has been a serious problem on ALL previous versions. Fixing this leaves only PDF export on the long term 'to do' list.
- Text trace width can now be resized. Broken since version 6.00!!!
- Default new label vertical spacing reduced from 40 to 35 mil.
- New label now positioned beneath the current label.
- Layer on/off status is now stored in the project file.

v7.16

- Block elements are all increased in size by 1 mil to ensure overlap when creating export files.
- Export readme files now include the statement: 'All Files Viewed from Top'
- PDF export now supported.
- A3 size paper now supported for postscript and PDF export.

<http://www.dilwyn.me.uk/graphics/index.html>

Double Apology

QL Today owes Dilwyn Jones a double apology. Two issues ago we spelt his name in three different ways of which, obviously, two were incorrect. To our great shame one of these appeared on the front cover and another in the index.

To compound our transgressions the editor failed to include an intended apology in the last issue.

Ruth Fegley

Herbert Schaaf has announced the death of Ruth Fegley who was an active USA QL enthusiast and leader in the CATS computer group.

An obituary in the Baltimore Sun described her as being "a leading force for civil and women's rights in the City of Baltimore ever since coming to that city in 1958. She conducted many programs to promote better interracial and interfaith relations in the City of Baltimore, with a

special emphasis on art and music activities for young people."

Tony Firshman published a link to the obituary: <http://www.legacy.com/obituaries/baltimoresun/obituary.aspx?n=ruth-fegley&pid=143725240>

Her family have set up a memorial website at: www.RuthFegley.com

Ruth Fegley



10/21/1923 - 05/27/2010

Nodding Donkeys, Steam, Air, Wind and Oil

by Tony Firshman

When I heard about the QL show in Austria I knew I had to go. I stopped active QL work a long time ago, but the friendships continue.

I was not nearly as efficient as Jochen and paid almost twice the amount he paid for a small twin bedded room. He had managed to book a vast apartment large enough to house a QL show! I am sure there was an undiscovered kitchen!

At the last minute both Marcel Kilgus and Per Witte decided to come. Unknown to each, these was some symbiosis. Both were in two minds about coming, but Per's decision to come encouraged Marcel. For a while the plan was for Per to sleep on the floor, but he won the Danish lottery ("a small win only") and had a smoking room of his own.

A bit of serendipity. Per sent me two Compswitches many years ago for repair. I totally forgot about these, but stumbled across them about the time Per said he was coming. He got his two repaired power strips and I did not need to cash his two year old postage cheque!

Anyway I booked my Easyjet flight from distant Gatwick (air). No problem as they still park motor-bikes for free, as do most other UK airports. When Gatwick was sold by BAA, there was a

move by the new owners to charge, which fortunately failed. I checked the website and they say bikes park for free in "Car Park 2". No sign of that description anywhere else on the site, especially the maps. Only 'Short term' (which is free in Luton) 'long term' 'extra long term' and so on. It took a week and much emailing, for them to finally confirm it was short term. Idiotic.

I arrived uneventfully at the Holiday Inn, Vienna at about midday. Marcel, Jochen and Andrea had already arrived. Per was due later in the evening. I had my first bath of the day (steam) and Marcel and I went off on a tourist run to Schönbrunn (Franz Joseph's palace). At first site it appeared to be held up with Grecian columns. He must have been short of money as most are fake! In front there was a wind band rehearsal (wind). Only when we got closer did we realise there were maybe 500 players, most dressed up in colourful outfits with Tyrolean hats.

<http://tinyurl.com/34c4uhu>

This was part of a weekend festival to celebrate 50th anniversary of the Vienna Wind Music Association. On walking round the back we saw two giant generators being moved. This turned out not to be for the wind band but for a

forthcoming outdoor orchestral concert. There we managed to photograph the rear end only of a visiting Czech cheer leader group (**steam** from both our ears). The lighting technology in use for the the outdoor concert was truly staggering. There were a great number of 125kW diesel generators scattered around the gardens (**oil**). Just one of these sites had ten 7kW "Space Cannon" arc spotlights, each one controllable wirelessly and connected via a control box using serial data links (XLR connectors). The main power cable from the generator was maybe 50mm diameter. Think 20th Century Fox and you will know what they were being used for: <http://tinyurl.com/37rseps>. These were on top of Gloriette - a folly on a hill overlooking the palace and Vienna. Both of us were struck on how like sewage farms the two 'lakes' looked, and very few ducks. We became **real** tourists for a short while as we drank coke (Marcel) and a mightily expensive coffee (me).

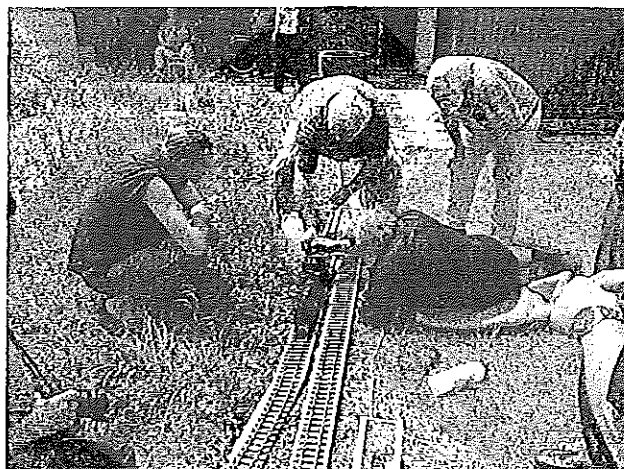
In the evening we went the the Prater Amusement Park. Remember the ferris wheel in *The Third Man* film? It is there still and looks amazing. I was disappointed though not to hear zither music anywhere in Vienna (8-)#. Jochen and Marcel went up what must be the highest roundabout in Europe - wide grins from Jochen especially. We then ate almost underneath it. Andrea and I shared the 'famous' pig's thigh - gigantic and delicious, especially the crackling.

The following morning we all drove to Prottes for the second day of the show. Nearing the show, we were struck by the smell. All became clear when we saw dozens of oil pumps (**Nodding donkeys, oil and gas**). It is Central Europe's largest oil field (<http://tinyurl.com/32h799f>). Ironically the area was also home to hundreds of wind generators (**wind**). The cynics amongst us said they were to drive the motors on the oil pumps.

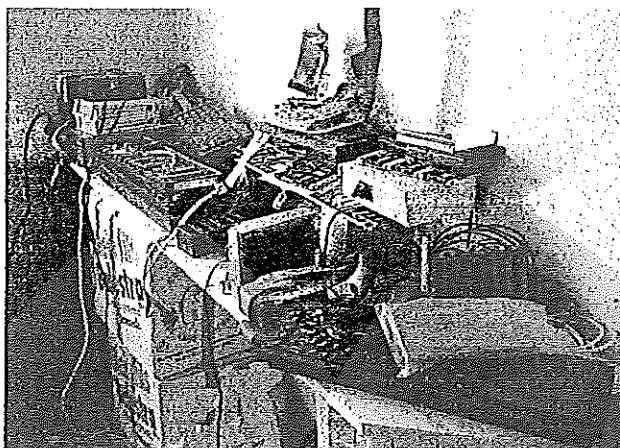


The show was held at Gerhard Plavec's newly purchased retirement home.

The grass had been especially cut for the occasion, but a 'forest' of grass was left in the middle of a circular large gauge model railway track. A scale model steam engine was put through it's paces by the builder.



Every room was completely full of toys from computers to Lego. except the bathroom and toilet, but they had computer mags!

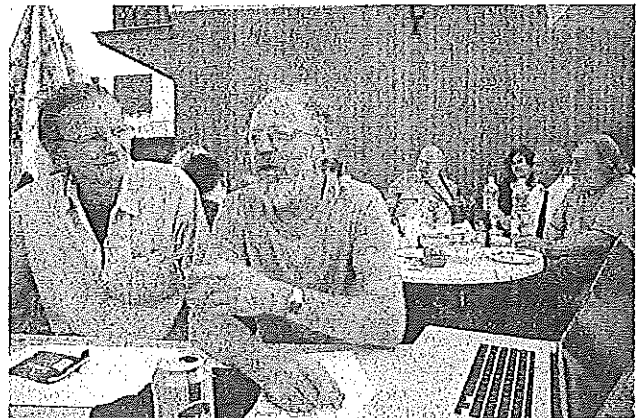
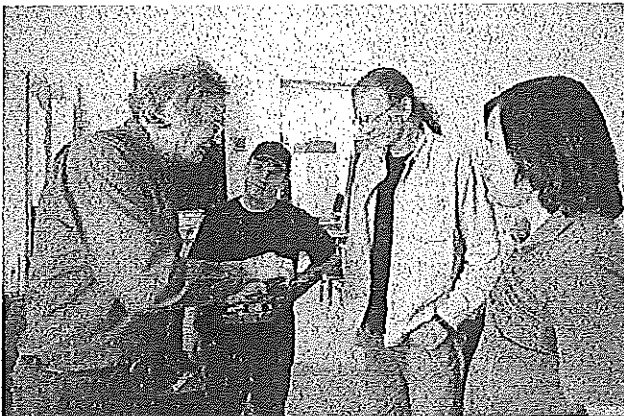


Urs König was mentioning how he would like a device to plug into a QL that would contain loadable software, but not a ROM. I mentioned USBWIZ. This is a serial device with on-board support for SD cards (FAT formatted) and some USB devices (Mice, printers etc). Adrian Ives is

developing it, and has working drivers. We have not yet tried with superHermes, but that should allow 230400 bps. However this would need a modified QL, with Hermes at the minimum for reliable use (at 19200 max) and power. I also talked about my Compukit UK101 that needed a huge voltage regulator. Now Gerhard's house must have **everything** as I found an example of the power transistor-like package used by the voltage regulator, and also a Usbwiz!

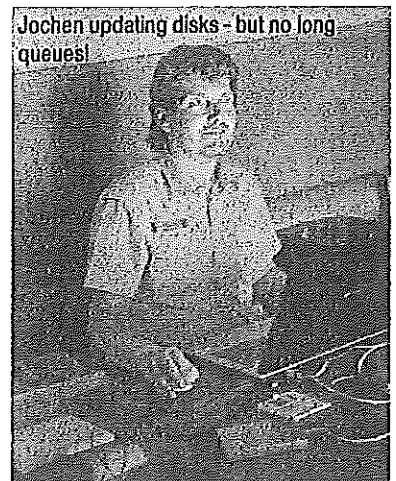
A excellent BBQ lunch was provided by the organisers with loads of beer!

It was good to see a French contingent (Tony Tebby/Wolfgang Lernerz), neither of whom are French of course, and partners. It was the first time I had met Beverley Tebby, and was delighted to hear she sings in a French choir. Most times on trips I suffer from withdrawal symptoms. A memorable exception was the last US QL show when I sung in Mary Boyle's (Bill Cable's wife) church choir.



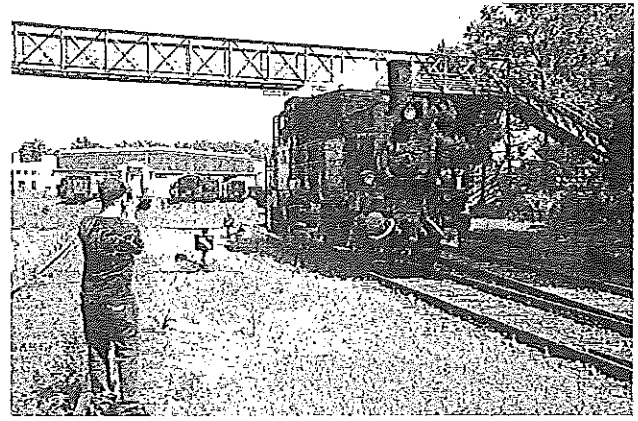
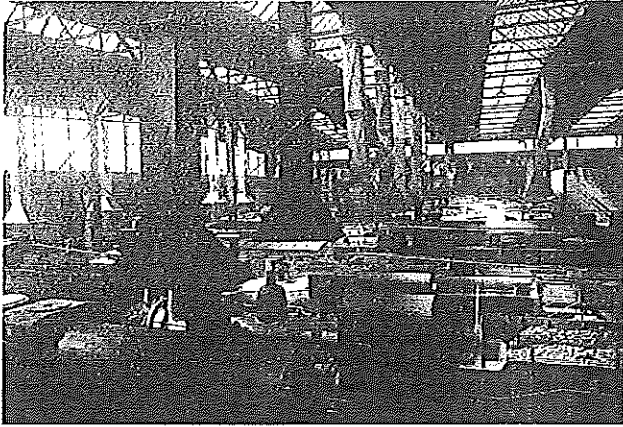
THAT must have been a very enlightening chat with Urs, Marcel etc.

We had 56k internet connection courtesy of Jochen, and I sent an email for Tony Tebby in German to a French speaking friend of his, enclosing a picture of the Austrian opera house (Volksooper Wien) they had been to the previous evening. They were off to St Stephen's, Vienna on Sunday for a Mozart Coronation Mass concert. I have sung in it many times, and would have loved to go, but there was another show day and a train museum (steam).

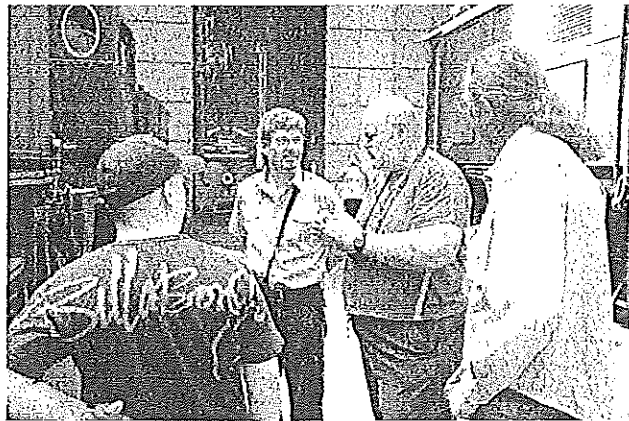


Jochen updating disks - but no long queues!

.... so to the third day. This was devoted to a visit to the Eisenbahnmuseum Strasshof - a railway museum (**steam** and **oil**). This was a superb



sion was the great working smell of the vast engine shed. There was no diesel or coal smell - only **oil**. The other souvenirs (oil and grease marks) have now washed off unfortunately. You can't get either of those experiences from the internet (yet)!



Back in Vienna on this last day (for Marcel and I) we ate at a restaurant near St Stephen's which was covered in posters for the mass I missed. I at last got my schnitzel, and the "Figmüller" was the largest I have ever eaten, or seen for that matter (<http://tinyurl.com/2vgn5a>)

Marcel and I were flying about the same time, so we took the fast, quiet CAT train to the airport. Easyjet completed a flawless double, and so I arrived home before midnight.

museum at all levels. The exhibits in most museums (like Verkehrshaus Luzern at QL-Mac) are perfect but display only. All the trains there, except in a controlled workshop area, were free to climb on. A lot were working and clearly recently driven. One was steamed up and offering short rides. There were also a lot of passenger carriage wrecks, often full of car parts! There were also model trains, and a slightly larger model steam train offering rides. It was a commercially run site, but seemed to be manned entirely by enthusiasts, most of whom were probably not paid. The shop keeper even asked for my camera and took pictures of us all. The lasting impres-



Another great QL holiday and I look forward to the next. More photos can be found at <http://ifs.firshman.co.uk/ql/photos.htm>



Klein Schiffi, who is to be seen with Jochen and Andrea in all the best places in Europe, gets pride of place on a cleaned bit of a giant oily locomotive.

Goodbye Paper?

by Geoff Wicks

Recently the question of an electronic version of QL Today has again been raised. Three years ago we explained why we had no plans for this. Since then we have followed the progress of Quanta's electronic magazine, and this has reinforced our feeling that electronic publication is not yet for QL Today.

When Quanta first produced its electronic magazine many members were still using slow dial-up internet access and Quanta restricted the file size to 0.5 Mb. This meant that there was no place for display advertising and limited opportunity for photographs and illustrations. Eventually Quanta realised it had set the file limit too low and raised this to 1 Mb. More recently those members of Quanta who have opted for the electronic magazine have indicated they would prefer a higher resolution and more colour giving a file size of 2.5Mb.

Our problem at QL Today is that an individual issue of the magazine is larger than an individual issue of the Quanta Magazine and, even in the most ideal situation, we would have a file size well in excess of 5 Mb. As we explained 3 years ago we do not have an ideal situation.

QL Today is produced with Calamus, which is an emulated ATARI program that does not use PC fonts. The PDF file has to be written as a bit map pixel by pixel and this produces a huge file size. As an experiment we produced two issues of volume 10 as lower resolution PDF files but the quality was disappointing.

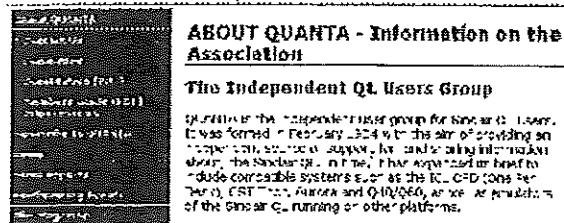
The proof-read copies of the magazine are produced as low resolution PDF files and twenty pages of the last issue had a file size of 1.8Mb. On page 7 of the issue we printed a screen shot of Quanta's website. Our illustrations show this image in the proof read copy and as it appeared in the magazine. You can imagine how high the file size would have to be to give a reasonable reproduction of that image.

It would, of course, be possible for QL Today to go over to other production techniques, but this would involve a huge investment in time to learn to produce a magazine to its present technical standards. The evidence from Quanta is that the take up of an electronic version would be too low to justify this expenditure of time. Quanta has 177 members, but only 30 have opted to receive the magazine electronically. Members have indicated that they prefer a paper version.

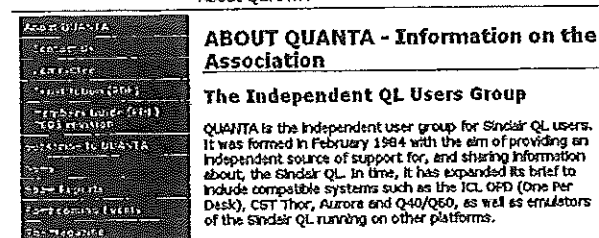
Those people who would like to see an electronic QL Today are not concerned about the file size. Three years ago a reader, who was on pay as you go dial up, indicated that he would not object to a file size of 10Mb and more recently a reader on broadband has indicated a download size of 100Mb would not worry him. I can understand the latter because twice a year I happily download a railway timetable of almost 4000 pages weighing in at about 50Mb, but file sizes of this size would put many other people off. It would also give QL Today a technological headache to work with download files of this size, as I discovered earlier this year when I tried to email some 15Mb of photos to the Quanta Magazine.

This reader pointed out that portable reading devices are increasing in popularity and this is something we shall have to bear in mind for the future.

My personal opinions differ a little from my opinions as editor of QL Today. I believe the days of paper QL publications are limited and that the future lies in a stronger QL web presence. However this would not be in the form of an electronic magazine. I hope to write something about this for the next issue.



Low-resolution example



Higher resolution example

1992 - Microsoft Windows

The first three versions of Windows marked a gradual drift from the 'add on windowing' of the Atari ST, the Amiga and X, towards integrated windowing as found on the Apple Mac.

Windows 1, 2, 3

Version 1 was started before Domesdos, although it was not released until 1985. It had very little success. Version 2 was a minor upgrade, with hardly any more success. Version 3, grafted on MSDOS, was somewhat better, but it was not until Microsoft released Version 3.1 in 1992, with TrueType fonts, that Microsoft had a competitive GUI with scalable, proportionally spaced fonts, virtual memory and cooperative multitasking, slightly after Apple released system 7 for the Mac with a similar specification. Microsoft has often been accused of being the king of bloatware, but Windows 3.1 worked quite happily on a machine with 2 MByte of RAM - only 3 times larger than a fully expanded QL.

Windows network

Microsoft SMB (Server Message Block) is a peer-to-peer networking system that was built into Windows for Workgroups in 1992 to provide shared access to files and printers. Some hardy QL types might remember the TK2 network for the QL: although the raw performance was rather limited (about 100 times slower than the Ethernet of the period) by the network hardware which comprised a one bit software driven I/O port without even an interrupt facility, it did actually work, serving files over the network, sharing printers, even providing clunky instant messaging. The QL network was derided, not because it was slow (and because of the extreme hardware limitations it was sometimes slower than Novell on a PC with Ethernet) but because it was peer-to-peer which was considered a poor relation of the established Unix dedicated server architecture.

Microsoft, bless their hearts, had realised that the dedicated server approach was a pretty stupid solution for ordinary office requirements and provided a peer-to-peer system that worked fairly well and still does - moreover, like the TK2 network, it does not preclude the use of dedicated servers, so you get the best of all worlds.

SMB is the most important Microsoft technology to be built into Linux.

Windows NT 3

Windows NT must be considered to be the sort of brainstorm error that can only be made by companies that have entirely lost contact with reality.

When it was released in 1993, Bill Gates said that NT stood for New Technology, but, although it did incorporate some peripheral technology from the early 1970s (Internet Protocol, Alto-derived GUI), for the rest it was just another system based on recycled 1960s theories. Other explanations given for NT is that Windows NT started off life as WNT (one letter up from VMS Dave Cutler's system for the VAX) or N-Ten - the development system used.

Is Windows NT really different from Unix or is it just another flavour of Unix? The parentage is in no doubt. Both NT and Unix are descended from Multics: Unix via UNICS, NT via OS2 on one side and via VAX/VMS and Mica on the other. Mica was designed to be Unix (BSD and Posix) compatible, NT was designed to 'resemble Unix'.

The defining differences between Multics and UNICS were the elimination of multiprocessor support, the radically simplified multi user security model and the hierarchical filing system. As Unix developed, the multiprocessor support was restored and the 'process / thread dichotomy' (to use the Plan 9's developers' polite term) was added. NT has the hierarchical filing system and process / thread dichotomy of Unix with a slightly improved multi user security model which is, however, still closer to Unix than to MULTICS.

Even Microsoft's senior management is unsure whether NT aims to be better than Unix or a better Unix: 'The day I come in front of a Gartner audience and say I have a better Unix than Linux, that'll be a good day,' (Microsoft's CEO Steve Ballmer, October 20, 2005).

So is NT is a bastardised Unix, an evolved Unix; or a failed Unix copy? It was intended to be POSIX compatible (i.e. more Unix than Unix) and Microsoft's detailed list of differences between NT and mainstream Unix is only two pages long (which is a bit shorter than the 3000 pages describing one minor Linux revision).

How was it received? Very badly! The workstation versions were targeted as business use: they were just a bad joke. The server versions did not make much of inroads into the Unix dominated market.

1994 - Early Linux development

Although the first full version dates from 1994, Linux version 0.01 was released in September 1991; it had 10,239 lines of code.

Linux 0

It is widely reported that Linux grew out of Minix, but scratching around, I can find little evidence for this although there was some borrowing. Minix was created by Prof. A Tanenbaum, a fanatical believer in semaphores, microkernels and 'burying interrupt handling as deeply as possible in the operating system'.

Linux was much more like the original UNICS: Linus Torvalds defended the monolithic kernel vigorously against attacks by Prof. Tanenbaum, he had much to say about the mental health of anyone who suggested improving Linux by using semaphores and he wrote, right at the start, 'I also happen to LIKE interrupts, so interrupts are handled without trying to hide the reason behind them' (comp.os.minix).

Setting aside the minor fact that in the ensuing (very public) argument Prof. Tanenbaum was provably wrong on almost every count, Linus Torvalds' approach was vindicated when Linux started displacing 'advanced' versions of Unix, with their semaphores, microkernels and the rest of the 1960s theoretical junk, simply on the basis that Linux worked better – as should have been expected by anyone who bothered to do a proper theoretical analysis (or by anyone who really understood computer systems).

Linux 1

Version 1, including the X Window System, was the first complete Linux version, which appeared in 1994, 2 years later than announced.

Although it was free, the take-up was minimal outside the academic world. It did not usually work when installed, even if the user managed to install it. The shortage of device drivers meant that Linux could only be used with very basic PC configurations. The situation improved gradually over the next decade.

1995 to 2000 – Wirth's law and its proof

1995 - Wirth's Law

'Software is getting slower more rapidly than hardware becomes faster'¹⁷

This is the core of Niklaus Wirth's Plea for Lean Software. Unfortunately it really comes out as a lament for his Oberon operating system and programming language. In the same way as Richard Gabriel could only explain the general preference for Unix and C over his operating system and language by resorting to the 'Worse is Better' proposition, Niklaus Wirth could not understand why the world seemed to prefer bloated software to his neat, compact operating system and language.

The 'law' was based on a comment in Martin Reiser's preface to the Oberon System Manual, 'The hope is that the progress in hardware will cure all software ills. However, a critical observer may observe that software manages to outgrow hardware in size and sluggishness'. As this was written at the latest in 1991, it can only refer to mainstream Unix, not, as many people have claimed, Windows NT.

¹⁷ A Plea for Lean Software, Computer, vol. 28, no. 2, pp. 64-68, Feb. 1995

Unfortunately, in the IT world, only cranks and those living in the past do not know that it does not matter how inefficiently software is written because you can always get a more powerful computer. He is reputed to have said with self deprecating humour "Whereas Europeans generally pronounce my name the right way ('Nicklouse Veert'), Americans invariably mangle it into 'Nickel's Worth'. This is to say that Europeans call me by name, but Americans call me by value".

If, by this comment he meant to imply that his ideas on software neatness were not appreciated the other side of the Atlantic, the USA / Europe software divide has no more value than Richard Gabriel's MIT / Bell labs divide. Europeans are as much responsible for bloat as our American cousins. On the other hand, it might have been a joke about parameter passing.

1995 - Microsoft Windows Grows

With the failure of NT 3 to achieve any significant sales, part of Microsoft pressed on with making products to sell.

Windows 95

Windows 95, released in 1995, was radically different from the 1,2,3 line, but was compromised to maintain compatibility with the earlier versions. Although the only "headline" architectural difference between Windows 95 and Windows 3.1 was the "pre-emptive multitasking", the system was much better integrated and the windowing system was WIN32: Windows 3 that had been re-written for NT 3 and then carried back to Microsoft's home operating system.

It also marked the start of Microsoft bloatware: although it was not in the same league as Unix bloatware (requiring about a fifth of the RAM and a quarter of the processor speed of Unix / X for an equivalent workload), it did gobble up about 4 times as much RAM as Windows 3.1 (for a very small increase in functionality).

The taskbar (which had appeared on the QL as the rather primitive "Button bar" (1986), re-cycled as the Acorn Arthur "Icon bar" (1987), the NextStep "Dock" (1989) and later patented by Apple in 2008) made its first appearance on a mainstream computer system. Unlike most other manifestations, it was, in true Microsoft "hedging-their-bets" style, not a replacement for the desktop icon mess, but an adjunct. It might not have been very original, but it was quite well done.

Windows NT 4

The best thing about Windows 95 was that it was not Unix. The Unix clan within Microsoft, however did not give up and they had support right from the top. In 1996, Microsoft launched Windows NT 4. This inherited the Windows 95 user interface and it had Microsoft's Internet Information Server as well as the SMB peer-to-peer network. For servers, the rapidly increasing computer RAM sizes and processor speeds meant that the stunning inefficiency of NT was becoming less of a problem. IIS proved to be competitive with mainstream Unix internet protocol servers (and as vulnerable) and in combination with SMB for local file serving, NT 4, with its far higher level of integration and coherence than any Unix system, started to be adopted for "enterprise servers". It was significantly helped by the inability of the Unix world to get to grips with the mass changeover, in the 1970s, to displays and keyboards with both upper and lower case letters.

As a workstation operating system, NT 4 was a disaster. Its customers, both business and personal, preferred Windows 95 and Microsoft was forced to continue supporting it and released Windows 98 as an upgrade.

Windows 98

Windows 98 and 98SE provided very minor upgrades (USB, Internet connection sharing) and fixed some problems at quite a high cost: the recommended RAM size for 98SE was three times greater than for Windows 95 – Wirth's law in action – welcome to bloatware.

For a while Microsoft seemed to be the unlikely saviour of the world as the last bastion against the encroaching Unix hordes. Unfortunately, the Unix rot was too pervasive within Microsoft.

Windows NT 5

Microsoft upgraded NT to NT 5 and tried branding it as Windows 2000, but very few people were taken in. All that NT offered them over Windows 98 was lower performance, an apparently insatiable demand for memory and seriously difficult system maintenance – who would want that?

Forced into either continuing to support Windows 98 or upgrading it, Microsoft decided to make a sort of hybrid "Millennium edition". Disaster again.

1996 - Linux 2, a radical shift

Linux 2, released in 1996 marked a major shift towards the academic world and the mainstream Unix server market with support for shared memory symmetric multiprocessing; first with a Big Kernel Lock and then with finer grained locking. It also marked a major shift in policy. In the early days, Linus Torvalds had vigorously opposed not only locking, but also making separate versions of Linux for different applications. Linux 2 could be compiled with and without symmetric multiprocessing support.

2001 to 2005 – Steps back and forwards

2001 - Microsoft stabs its own customers in the back

Microsoft's customers had made it very clear what they thought about Microsoft's Unix-like NT operating system, but Microsoft did not seem to consider that the NT development policy could possibly have been misguided.

Microsoft's recovery plan for the Millennium edition mess was to rush out a minor upgrade to NT5 (NT 5.1) in 2001, branding it as Windows XP with a massive publicity campaign, this time targeting their core market, small users. The new version of NT had both "home" and "professional" editions. Realising that the customer resistance to Unix levels of performance would have to be overcome, nothing was left to chance. On the one side, Microsoft advertised the "power" (i.e. inefficiency and insatiable hunger for memory: remember the "powerful" Amiga operating system) of XP – the recommended minimum configuration was 8 times more RAM (128k vs 16k) and two generations of processor (Pentium II vs i486) by comparison with Windows 98. XP was, therefore, destined exclusively for a new generation of "Designed for Microsoft Windows XP" labelled PCs: there was no question of running it on a one-year-old machine. On the other side, Microsoft withdrew Millennium edition and announced the imminent withdrawal of support for Windows 98. As a result, XP was widely, if reluctantly, adopted for new machines. After all, what alternative did users have? Linux?

2001 - Linux 2 grows

Over the years, Linus Torvalds seems to have lost his grip on Linux.

Kernel version 0.01 had 10,239 lines of C (against Domesdos's 5,000 lines of assembler), kernel version 2.6.30 had 11,637,137 lines of C. Did it really have a thousand times the functionality of version 0.01?

Not only is the code size of version 2 ballooning for little visible improvement, with the recent 2.6.26-rc1 kernel, the AIM benchmark ran 40% slower than with the previous release¹⁸. The problem was tracked down to just one semaphore (how on earth did that get there?).

The problem was caused by removing 7000 lines of "unlamented" code from the general semaphore. Semaphores are usually treated academically as having zero cost because the cost cannot be quantified, but to streamline this semaphore, someone had removed about four times as much code as the whole of Domesdos and all its device drivers! Unfortunately the 7000 lines of "unlamented" code proved to have been necessary.

Linux is notorious for its hundreds of changes between its frequent releases, but these are mostly minor. I could well be wrong, but I can find only two changes that mark significant shifts in capability.

In 2001 Linux 2.4 introduced support for USB.

This was three years after Windows and Mac OS, which shows one of the limitations of the Linux development method. For the first time, Linux became a contender for PCs and workstations.

¹⁸ <http://www.linuxworld.com/news/2008/052008-kernel.html>

By 2004, various patches had appear to produce a real time version of Linux.

These were supposed to be included in the Kernel in later versions of Linux 2.6, but at the time of writing in 2009, the "real time" aspect of Linux performance seems to depend more on brute processor force and vain hope than on any real design to meet time constraints.

2002 - Mac OS X

In 2002 Apple withdrew the long running original classic Mac OS series in favour of the Unix based Mac OS X that had been previewed since 1999. There was now no non-Unix system in the mainstream personal computer and workstation market.

To compensate for this major step backwards, Mac OS X introduced a number of small but significant, steps forward.

For the first time on any mainstream system, the windowing was based on off-screen buffers (c.f. Pointer Interface for QDOS, 1986). The display was kept up to date, not by the applications, but by a background task, the Compositor, copying from the off-screen buffers to the display frame buffer. This was based on the same techniques as the various patches to the QJump extended environment (PEX, PNICE, PIE) in use from the early 1990s. As Apple was supposed to have control over its own hardware platforms, it is astounding that this was not handled directly by new display hardware with the separate window buffers being displayed directly. In fact, Apple no longer had control over its hardware platforms for workstations: it was now in the business of packaging and branding standard PC hardware – how the proud are fallen. A very clever solution was found by treating the window buffers as large 3D textures which could be painted by games-oriented graphic processors in the display controllers. "Clever" is one of the worst insults in systems development and maintenance.

The compositor also introduced drop shadows on windows (familiar to old QLers) had a patented "dock" (QPac2 1986) and allowed for translucent windows (novel and pretty, but still looking for a useful application nearly 10 years later).

2005 and a bit - Windows NT 6

Windows Vista (NT 6) was announced in July 2005, but was not released until more than a year later. However, chronologically, it belongs here. Packed full of features to make it faster and easier to use than earlier versions, it was, as a result, enormously larger, slower, less coherent and more difficult to use.

Vista take-up was effectively limited to those who did not know how to avoid it - three years after it was released, it had only reached a penetration of 25-30% of the Windows base. If Vista had been no worse than XP normal replacement, new sales and piracy should have pushed it to well over 70% after three years.

The most visible feature of Vista was the windowing. Best considered as a copy of Mac OS X Quartz, the new Desktop Window Manager (DWM), based on off-screen buffers, showed "none of the redraw artefacts, latency, or tearing effects that you may encounter in earlier versions". "With the Desktop Window Manager, applications do not draw directly to the video memory; instead, they draw their contents to off-screen buffers in system memory that are then composited together by DWM to render the final screen, a number of times per second".

Microsoft developer Greg Schechter¹⁹ explained the significance; "when a window moves across the screen in XP and before, the portions of background windows that are newly visible only get painted when the background application wakes up and starts painting ... For non-responsive background applications, or even responsive ones that happen to be paged out, this can yield a very poor user experience."

In 2005, this had been known for decades, but the real significance is that although the "user experience" is very poor even if all of the applications' display data structures are "100% correct, 100% of the time", there is enormous cost and difficulty in ensuring that this is true – another major contribution to software unreliability and bloatware – so why was this technique ever used?

Vista also introduced another "new feature" just after it appeared in mainstream Unix, Linux and Mac OS X: Address Space Layout Randomisation (ASLR).

19 http://blogs.msdn.com/greg_schechter/archive/2006/03/05/544314.aspx

Box 11 - The Unix buffer overrun vulnerability

Why Unix in particular?

Because the Unix environment (which includes C compiled executables) brings together three "design choices", each of which should probably have been avoided for various other reasons, that create a unique vulnerability.

1. Fixed address, virtual machine model of execution.
2. Intrinsically unbounded string data structure.
3. Variable length data storage on the procedure return address stack.

Together, these make it possible for an outside agent (another computer on an internal or external network, for example) to send the target computer data which will overflow a string buffer on the stack and overwrite the procedure return address. The data overwriting the return address defines the point in memory where the execution continues when the current procedure returns. This could be code within the program or a dynamic library or code "injected" into a string buffer on the stack or elsewhere in memory.

1 Fixed address, virtual machine model of execution

This model was the basis of the multi-user MULTICS system. The intention was to create a virtual machine for each user or "process", so that, each time a program executed, on any computer, the environment (including all addresses) would be identical. Each process, therefore, had its own fixed address space that duplicated the address space of every other process.

This was not the execution model used by most conventional multitasking operating systems of the time, where all programs shared the same address space: every time a program was executed, it could, potentially, be loaded at a different address in memory. This meant every program had to be either "position independent" (this is explicit in the MC68000 instruction set) to execute at any address or "relocatable" to execute at the address where it was loaded. The shared address space execution model has the advantages of being simpler for the operating system, potentially cheaper in hardware terms and providing a much more efficient operating system interface. These are very good reasons for not choosing a fixed address execution model, even if you are not worried about vulnerability.

If the address at which a program executes is fixed, then it is possible to have predictable results if a subroutine return address is overwritten with any one of a very large number of values. On the other hand, if the position at which a program executes is not fixed, then it is "practically" impossible to overwrite return addresses on the stack and have any sort of predictable result.

2 Intrinsically unbounded string data structure

Although string variables are not intrinsically defined by the C language, string constants are. As a result, the whole of the C environment assumes that string variables use the same structure as string constants. This structure defines neither the length of a string nor the length of the buffer in which a string is stored: the end of the string is marked by a sentinel and the end of the buffer is not marked at all.

This C structure is less efficient than having defined length strings for any of the range of string operations that are commonly used other than copying or concatenating short strings. Furthermore copying a short C string is only significantly more efficient for certain processor architectures and only if no check is made for buffer overrun.

To make it worse, variable length strings are not identified as such in the language, they are merely a convention so there can be no systematic checking. The C choice has been justified on the grounds of efficiency: an intrinsically inefficient string representation was chosen and so bounds checking could not be considered as this inefficient representation makes bounds checking very costly.

3 Variable length data storage on the procedure return address stack.

C is not the only language where local variables are stored on the return address stack, this is a natural consequence using a recursive programming language on a stack based processor. One of the 1960s programming language dogmas was that recursion was "elegant" and, therefore, should be considered to be the normal paradigm for subroutine calls rather than an exceptional or even aberrant method of handing nested data definitions.

For C and other languages of the period, this was not too much of a problem for fixed size data, but storing variable length data on the stack is, however, a different matter if you have no idea how large the data is going to be in advance.

C adds a twist to this: lengths of C strings are not just unknown in advance, but unknown even when they are being processed. The usual C design philosophy was to allocate a much larger buffer than usually necessary (very wasteful) in the knowledge that buffer overrun would be very unlikely unless you deliberately set out to do it. Who cares about people who deliberately set out to create buffer overruns?

2005 - Address space layout randomisation

The arrival of address space layout randomisation in mid 2005 (Mac OS X.5, June 2005; Linux 2.6.12, June 2005; Windows Vista, announced July 2005, but already longer in development than the others) is notable for the vast quantity of exaggerated claims, pseudo-mathematical nonsense and misleading information that has been generated about a dirty little patch for a problem that should never have existed.

The intrinsic vulnerability of the combination of the Unix virtual machine execution model and the C function call mechanism to buffer overrun attacks (see Box 11) had been known since the 1970s. The first major attack (the Morris Worm on BSD Unix) took place 3 years before the version 0 of Linux was developed, 5 years before Windows NT was released and 11 years before Mac OS X was previewed. Why is this vulnerability so important, why would anybody produce a new operating system with this known vulnerability and why did it take so long to patch it?

The importance of the vulnerability is twofold. Firstly, buffer overrun exploits can potentially infect Unix type computers at any point where they read data from the outside world, in particular via Internet protocols. New exploits are close to undetectable until after the computer has been infected and the damage has been done. Viruses, on the other hand, because they can only spread by the transfer of infected executable code from one machine to another, can be prevented by simple prophylaxis. Secondly, because the vulnerability exists at the level of normal Internet data transfers, rather than being limited to the installation of infected software, it creates the possibility of a computer being "taken over" and controlled "invisibly" by a remote system or user whether or not internal or external firewalls and anti-virus systems are used.

I do not think that Linux, Windows NT and Mac OS X were deliberately designed to be vulnerable. It seems more likely that their designers were suffering from the same tunnel vision as the designers of Plan 9, who, in the early 1990s, were able to write that the majority of personal computer workstations were running Unix when in the real world the usage of Unix was so low that it did not even figure in the charts. There was a whole generation of computer scientists who, although they may have had the occasional brush with "real world" operating systems, had learnt computing using Unix (the original open source operating system), had learnt Unix fundamentals as "universal truths" and simply did not know that the vulnerabilities they had learnt about were specific to Unix.

It took a long time before any patches were produced to deal with the vulnerability because it was not the result of a simple oversight: the problem was fundamental to the design of the Unix / C environment. It was so fundamental that, for the major weakness (the fixed load address for executable code), the patch for Windows was only effective for dynamic libraries and special executable programs marked as relocatable and not for existing programs, the patch for Mac OS X was ineffective for any executable program, while the patch for Linux ...

In the next issue, Tony explains the 2005ish situation and more recent developments.

Zoostorm Freedom XL 10-270

by Dilwyn Jones

For some time now, I've been using an eeePC netbook, running Windows XP as a kind of portable QL. My QL system these days consists of QPC2 running on any suitable Windows platform, so as a portable QL system, it was fine even if the 7 inch screen and tiny keyboard and slow 900MHz processor were a bit of a drag. That said, QPC2 ran fast enough for me, and if I really wanted to I could plug in a full size keyboard and external monitor.

Recently, with the decreasing costs of netbook systems, I decided to upgrade from the eeePC when I saw an Argos special offer on the Zoo-

storm Freedom XL 10-270 netbook, at just £199 (part number 508-3053 - I don't know when the special offer ends, though)

At the time of writing, I've been using this ultra-cheap 10 inch screen netbook for about 3 weeks and am really getting to like it! It's some way from being the most advanced Windows netbook or small laptop you can get, but at this price I'm not complaining as it runs QPC2 (and from what little I've run QemuLator on it, that seems to work fine too).

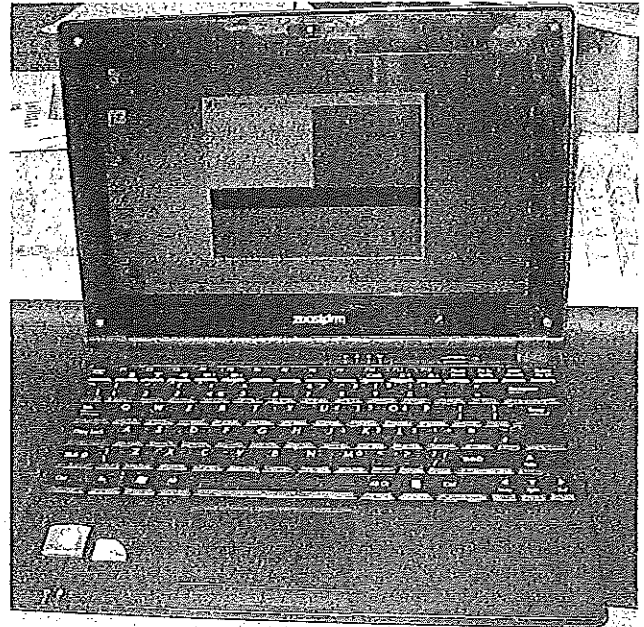
The spec is quite reasonable. It has good

storage - a 250GB eSata hard disk drive - and it comes with Windows 7 Starter Edition, which limits what a manufacturer can do with such machines. It seems that 250GB storage, 1GB RAM and so on are set by Microsoft as limits for Starter Edition. This edition of Windows 7 is a very cut down model with many features missing from the versions you'll find on mainstream desktop and laptop computers. But, for what I needed in a small netbook like this, I find it perfectly adequate for use as a portable QL with the added functions of Windows such as email and internet access. Its 1.6GHz processor seems to cope quite well with the Windows 7 and everything I asked of the little machine. It has a built in webcam and speakers (though these are tiny and tinny as you might expect in a netbook) along with sockets for headphones and microphone, so you can use a headset for audio or video calls if you wish. The computer has an external VGA socket, so you can plug in an external monitor if you wish. The built in screen does 1024x600 pixel resolution.

SPECS:

- Windows 7 Home Starter
- 10.1 inch 1024x600 TFT screen
- 1.6GHz processor
- 1GB RAM
- 250 GB hard disk
- SD card reader
- 802.11b/g/n wireless
- 10/100 wired LAN
- 0.3MP webcam
- stereo speakers and sockets for external mic and speaker
- 3 hours battery life
- 2 USB2.0 ports
- VGA out socket for external monitor
- Some fairly basic software:
- M\$ Works9, M\$Office trial version

Now for the negatives, and there aren't many: The wireless was a bit of a trial to begin with (read reviews on the web which says early models were actually shipped with wrong drivers for it, that they recommend you delete the supplied driver then let Windoze reinstall it), probably because of a peculiarity of the Orange Livebox wireless router we have here (loads of computers have had problems pairing with it), but once I persuaded it to pair up after several tries, it's been fine since. Wireless reception is good - picks up more signals around our street than any of the other computers in this house.



The netbook with a traditional keyboard in the background

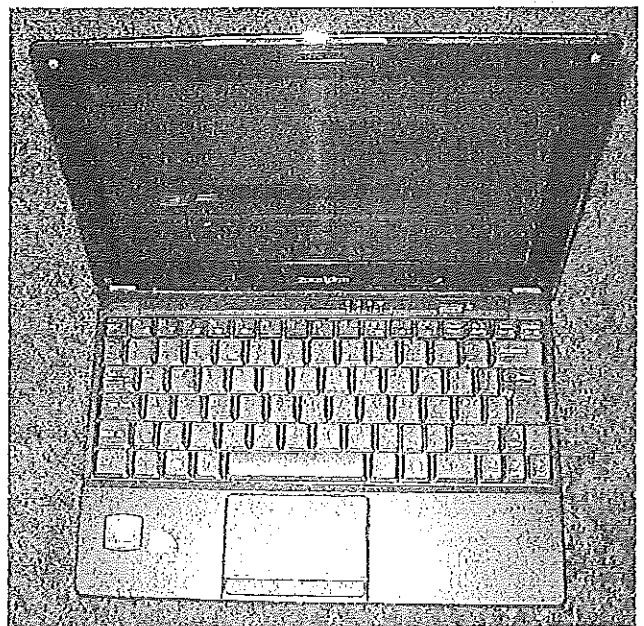
It has only 2 USB ports, though you can use a hub of course.

Battery life is a bit disappointing by modern standards at "just" 3 hours. You have to judge this against the cost of this system, of course.

Web reviews are mixed. People who like it praise it to the hilt, people who don't seem to be taking them back to Argos for refunds!

Technical Support is via Argos, not Zoostorm (though you can contact them via email via www.zoostorm.co.uk). Instructions are poor, one thin starter booklet only.

No system restore disk is supplied - make a backup ASAP which is not easy because you'll need an external USB CD or DVD writer or an external hard disk. No floppy drive either,



Keyboard overview

although it's easy and cheap to buy USB floppy disk drives which will plug and play as the usual PC drive A:

Other Opinions

I happened to mention this computer on the ql-users mailing list in July and got some mixed reactions.

Tony Firshman said:

"Indeed it is a pity about the battery life. Maybe with an SSD drive the life would be longer. The screens though are power hungry. Nothing will persuade me not to use my Apple Macbook Pro with 7 hours plus, other than a 12 hour life small netbook. The macbook does it partly by using LEDs for the screen rather than 110vAC strip lights. Maybe the IPAD when it becomes a real computer will do that, with Qemulator."

And Derek Stewart suggested:

"Get a ASUS eeePC, they are the same price and better battery life."

Conclusion:

For me, it makes a perfectly acceptable mobile QL with wireless for web/email access. Weighs less and needs less space than a conventional laptop, the 10 inch screen and almost full size keyboard is perfectly adequate. Windows 7 might be Home Starter edition, but adequate for the sort of uses I have in mind. 3 hour battery life could be better nowadays, but again adequate. 250GB hard disk is great, and very handy having a couple of USB ports and SD card reader. At just £199 with a cut down Windows 7 it's a good mobile solution for where a larger laptop is over-kill.

Even if you don't think much of this particular netbook, it has to be said that the increasingly good price/performance ratio of small netbook computers make them very attractive for the QL niche of emulator users like myself, while those who prefer to go the Mac route (like Tony Firshman) may be better served by waiting for the OSX version of QemuLator to come out later this year.

The Future of the QL

by Geoff Wicks

After weeks, if not months, in which the QL-users email group had appeared to become a technical helpline, it suddenly burst into life early in June with a major discussion on the future of the QL. Over a three week period almost 30 people participated in the discussion, some of whom declared themselves to be "normal list lurkers". QL Today's cut down version of the discussion that was used to prepare this report came to almost 24,000 words covering 67 A4 pages.

The discussion started with a contribution from **John Sadler**:

"The QL Community is steadily reducing in size. So what can we do to stop the QL disappearing into oblivion?"

Other old computers have survived by ensuring that emulators & programs exist in the public domain.

The QL does have public domain emulators but they are a hassle to use.

The problem is in either they do not work with all programs or else by the time the person has learnt to convert the program so that it will run they have already decided the QL is not for them. There is one emulator which overcomes this problem and that is QPC2. However when you have bought it you find you need QPCPrint to use a printer!

Then you find some of the best programs are commercial and that's more expensive.

George Gwilt suggested at the Quanta AGM that Quanta should buy the rights to QPC2 & QPCprint with its surplus cash so that they could be part of the public domain and people could appreciate the QL for nothing.

Unfortunately these seeds of wisdom fell on stony ground.

SO are you content to see the QL disappear into oblivion OR are you going to do something about it?"

QUANTA AND QPC2

Marcel Kilgus quickly responded that he had never heard of this proposal and indicated that QPCPrint is not for sale as its current target users are largely non QL-ers and businesses with old DOS software. However its functionality could be integrated directly into QPC2.

No response came from Quanta and only one member of the Quanta committee participated in the discussion and then in a private capacity.

QL Today can find only one Quanta reference to QPC2 in an AGM. This was in 2007 when the minutes record:

*A request has been made that a good use of QUANTA funds would be to give every member a copy of QPC2. This has been looked into with the following results:

The costs would amount to about £9000.00 which would wipe out any reserves QUANTA has.

The 2004 survey showed that 30% of our members already have QPC2.

It also showed that 30% only have a basic QL and therefore have no interest in QPC2.

It was therefore decided that this request was not a viable or required proposition.*

No reference was made to buying the rights to QPC2 and QPCPrint.

WIKI ETIQUETTE

Rich Mellor was the first person to give a detailed reaction to John Sadler.

"It does depend to some extent on resources, but also enthusiasm. From the last Quanta survey, it appeared that most of their members do not use emulators, but rely on original hardware. It will be interesting to see what sort of response levels they get to the latest survey and whether this has changed.

I would like to see Quanta work together with its members and the rest of the QL community to decide on how best to promote the QL and secure its continued future.

I have made a few suggestions in the past, but unfortunately, the Quanta committee did not appear to grasp the concept behind the ideas, or how they can help to promote the QL."

He then went onto to describe his Wiki experiences (http://www.rwapadventures.com/ql_wiki)

"There are several ideas behind this site, which work together:

a) To promote the QL, by increasing the amount of information available on the internet about its hardware, and software. b) To preserve copies of QL documentation and help to make them more widely available. c) To preserve copies of commercial QL software, so that if users buy a second hand piece of software, to find that the microdrives will no longer load, or if they have moved onto emulators, they can readily purchase a working copy (on production of proof that they own the original).

Unfortunately, despite my best efforts, the main contributors to the QL Wiki remain myself and Dilwyn Jones.

When I ask for comments / feedback or assistance, the only response appears to be "You have not included any information about [x]"

The QL Wiki is not designed to remain my site - it is for the whole of the QL community, yet why don't others pull their fingers out and contribute information to it / help to preserve images of software?

>From my own point of view, I am bound to concentrate on entering information which helps me to ensure the commercial viability of RWAP Software. This works and various people have come to my site as a result of the QL Wiki. However, none of the other traders or even Quanta have entered any information about their own products"

Both Marcel Kilgus and Jochen Merz said it was inappropriate to write about your own products, but ex-trader Tony Firshman had a more nuanced view.

Marcel wrote:

"In these cases I normally follow the rules for Wikipedia which basically say you shouldn't write about your own stuff but let others do/judge it. This is one of the reasons why I generally don't write articles concerning QPC etc."

To which Jochen added:

"I also feel when I put things about my own products into Wikis, it could be regarded as self-advertising."

Tony commented:

"You will find that most QL hardware and software is already documented, by Rich I think. I saw no harm in my going in and correcting what is there."

Rich Mellor then ventured a different opinion:

"1. Surely the software / hardware designers are best positioned to provide the background of the hardware and software? 2. The internet may be aimed at providing information, but 99% of sites (including wikis) are used for promotion of business - after all, not many people can afford to set up and maintain a website long-term without it providing some source of revenue. 3. What is wrong with self-promotion? Surely anyone who writes a CV out and puts it online on a job site is doing just that thing and nobody would argue against doing that."

Jochen further clarified his attitude:

"I regard any "encyclopaedia" (online or offline) as something which should be as unbiased as possible, and writing something about your own commercial product could not be regarded as this ... that's how I feel about it.

Please do not regard this as criticism to anybody who does so (Rich, for example, is doing a great job and I cannot imagine how much time

QUO VADIS
DESIGN

Independent Information
Technology Services

www.ql-qvd.com

QUO VADIS
DESIGN Independent Information
Technology Services

QL/QDOS/SMSQ/E Software

[Home](#) [Products](#) [Support](#) [Company](#) [Contact](#)

Welcome

Quo Vadis Design sells software for the Sinclair Quantum Leap computer (QL) and variants including a new OS called SMSQ/E.

The QL is a computer in its 25th year Anniversary.

The Sinclair QL - a quantum leap in personal computing



Software emulations of the QL now exist which can run on a PC/Mac with Windows/Linux or Mac Operating systems.

Copyright © 2009 Quo Vadis Design. All Rights Reserved.

[Home](#) | [Products](#) | [Support](#) | [Company](#) | [Contact](#) | [Privacy](#)

News

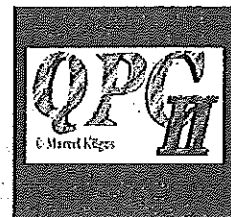
✂ QVD QL News Blog - keep up to date
[News Blog](#)

24/02/2009

✂ Quo Vadis Design Website Launched

01/02/2009

FEATURED PRODUCT



BUY NOW!

Bruce@ql-qvd.com

Quo Vadis Design
38 Derham Gardens
Upminster
RM14 3HA
UK

Tel: +44 (0)20 71930539
Fax: +44 (0)870 0568755

Special Offers available from
Jochen Merz Software for its
25 years in QL Trading

Check the QL News Blog on
our website for the special
offers



Subscriptions taken online

QL/QDOS/SMSQ/E Software

it takes him), but I would rather NOT do it for the products I sell.

It would, however, not worry me, for non-commercial products, as you do not (really) want to compete, so I would not regard this as advertising."

FEEDBACK

Norman Dunbar echoed Rich Mellor's experience of running a wiki. (<http://qdosmsq.dunbar-it.co.uk>)

Only he and **George Gwilt** had contributed to his wiki, although he has a number of readers.

He also expressed his frustration about the lack of feedback to his QL Today articles:

"My Articles in QL Today are a mystery to me. Not that I don't know what I'm writing (well, most of the time I do!) but the lack of feedback from almost everyone in the entire world - I sometimes feel that I'm writing for three people - Me, George and Hugh Rooms who called me to task some time back and made a contribution to the Assembly series.

Every time I ask for feedback, the silence is deafening."

Jochen Merz reported that for almost 10 years now it has been difficult to get any feedback for anything in QL Today.

(Although it was not part of the user group discussion Quanta has a similar problem. At this year's AGM QL Today had a long discussion with Quanta's chairwoman about how frustrating the lack of feedback is for all members of the Quanta Committee.)

TOLLS THE BIG BELL?

At this point in the discussion two ex-QL-ers described their experiences of returning to the scene.

Martin Wheatley wrote:

"I stopped using the QL a number of years ago and moved out of London. In an idle moment I wondered what was happening and rejoined the list.

The thing that is immediately worrying is that so far every one who has posted is someone whose name I know. If no one new and active has come in during the last few years then the big bell is tolling!"

Anton Preinsack described a different experience:

"I am kind of new here (although I owned a QL in the late 80s).

It started with a QL I bought on Ebay last year.

In the meantime I own a Q40, two QLs and use QPC2. The QL brought my interest for coding back. Its more fun to code for a simple (but elegant) system like the QL. So I train my skills in SuperBasic and Assembler at the moment and hope, that this ends in new programs for the QL from me.

So it is not impossible, that at least users, who owned a QL in the past come back. I also noticed, that there is a high interest on QLs on ebay.

What I miss (but what is understandable, because of the costs) is the lack of new hardware."

Rich Mellor echoed some of Anton's comments:

"Believe it or not, there is still a burgeoning market out there for retro computers - I rely on them for my main source of income at the moment (since I was made redundant in December) and have successfully brought several new products to market (or made old ones available again), and all this helps.

People do want new products, but they also want to be able to keep using their existing products, and that is why the software preservation side of the QL Wiki is equally important - many users find that as microdrive cartridges degrade, or they move onto an emulator, they no longer have access to their favourite programs, some of which are copy protected."

"List Lurker" **Jerry Davis** posted some interesting comments:

"I agree with Rich, there is an interest in retro computers. I am now an Engineering and Electronics teacher at GCSE and A Level. Electronics education has recently been revolutionised with the introduction of cheap PCs that can be programmed via a built in serial port. PCs are far too complicated for the average student to understand, however older computers such as the Sinclair computers are easier to understand and modify. While they are not QL specific projects, I have had GCSE students design keyboard interfaces and sound cards. I've just looked at the suggested projects for A Level next year, and one is to design an SD card reader - it doesn't specify what disc format the project should use.... In the not too distant future we will be looking at A Level students designing computers as complex as the original QL!

I think older computers such as the QL that are "less complicated", could be used to great potential in electronics education to get across the ideas of "Processor", "Math-Coprocessor", "Display Driver", "Keyboard interface", "disk interface" etc etc.

I dusted off my old QLs, and some students were amazed to see that you could network computers using just "iPod headphone leads".

THE UK SCENE

In a reply to Martin Wheatley Rich Mellor wrote:

"There are quite a few people who have come to the QL in the past couple of years - many of them as a result of my own website and purchasing second hand items offered by myself and/or Quanta. Some are more active than others.

However, I have a mailing list of over 1000 QL users and it is rare that any of them ask to be removed from the list.

So, the market is not quite ready for the death knell."

Geoff Wicks asked Rich if he could say how many people on his list lived in the UK and received the answer 523. Geoff then posed a series of questions:

"Rich has told us he has a list of 523 UK QL-ers and we also know how many UK Quanta members and how many UK QL Today readers there are. A little bit of arithmetic suggests the UK community is very different from how we envisage it.

Suppose we took Rich's list and removed from it all the Quanta members. We then went through it a second time and removed all the remaining QL Today readers. Finally we go through it a third time removing all the remaining subscribers to this list. After these three operations we have removed far less than half of the names on this list.

In other words if we divided the UK QL community into two groups that we could call the "organised group" and the "non-organised group", the non-organised group would be the larger of the two. It sounds almost unbelievable but the majority of UK QL-ers have no need for Quanta, QL Today or this list. There is a large part of the QL universe that is ripe for exploration.

Now three questions to think about:

1: Suppose someone from the non-organised group visited two QL websites - Dilwyn's and Quanta's. To which of the two websites would he be the more likely to return and why?

2: A couple of years ago two UK traders ceased active trading after making a loss for several years. A third trader remained and is running a successful QL/retro business. Over a 2 year period he also provided Quanta with a quarter of its income by trading on their behalf. How has he achieved this?

3: Quanta's recent survey was publicised by Quanta itself; on this list; in the news columns of QL Today; and to all people on Rich's list. Which of these four would provide the largest potential number of respondents?"

At about this point Roy Wood entered the discussion:

"Nothing has a future if it does not move. All the while I was writing 'Byts of Wood' I was up against the fact that there was nothing to write about. Nothing new. No point in being a trader for a static community if all you are advertising is the same stuff that everyone already owns. Then, when you come up with a great system to make things better, no one buys it (QDT of course). There may well be 500+ users on Rich's database but what are they doing with the QL? What has he sold them? The odd keyboard membrane, disk expansion? How many repeat sales? Data is only of use if it has a context.

In all the time I was a trader there were very few ground breakers and they gradually fell by the wayside through lack of support and sales. This has been the most active conversation on this list for ages and a while back there were people talking about how to print - a subject that has been round the track more times than a sprinter with Alzheimers. When the QL community was thriving it was moving forward with ideas flowing. These last years it has been inward looking and characterised by infighting and lack of inspiration. I have not seen QL Today since I stopped distributing it but I am still a member of Quanta and I see nothing new there. If you want it to continue then you have to stop complaining and waffling on this list and write programs, have ideas and innovate. No point in magazines and user groups if there is nothing new to say or do.

This may all seem a bit harsh but, like a jolt to the heart in case of cardiac arrest, sometimes you need a defibrillator to restore a pulse. The paddles are in your hands - don't wait for the flat line."

Phil Kett wrote a lengthy reaction to Roy's contribution:

"If you look at the communities surrounding the old computers, whether they be sinclair, commodore, atari or whatever - in nearly all cases it's the games that keep the computers alive. Yes, there are hardware innovations, but it's nearly always the ability to load games quicker that prompts the development - the divide interface for the spectrum springs to mind.

People unfamiliar with the QL aren't going to want to try and use a word processor from the 80s when the one installed on their PC is 100 times better.

The availability of software is also something that isn't there in the QL community. Search for just about any other 80s computer and you'll find a wealth of software available on the net. The legality of these downloads may be suspect but they're there - search for QL software and what do you find? One or two sites offering compilers and productivity software or asking you to purchase games.

The QL is a fairly obscure platform and no one that is unfamiliar with it is going to pay money for something on the off chance that it might be good.

There is another problem with QL software - it's fragmented. Some software will run on a basic QL for which you can download an emulator. Other software requires an enhanced system for which you need to either have the hardware to run it or purchase an emulator. For just about every other 80s or 90s computer you can download an emulator that will run all the software for that machine.

I think we need to face the fact that the QL is almost a dead machine. People aren't willing to invest money in something if they have no compelling reason to use it and let's face it most people don't even know what the QL is!

There are thriving home-brew communities for a lot of the old computers but the QL isn't one of them. Why is this? It's not a money thing, a lot of home-brew software is given away free. Basically there aren't that many people who are bothered about the QL - why write something for the QL when you could write something for the Spectrum and get a wider audience?

The QL is a dwindling niche market and while people continue charging not inconsiderable sums for software it will remain that way (until it dwindles into oblivion)."

Over several emails **Dilwyn Jones** made a number of comments about Roy's contribution. Firstly he commented about the problems of the Quanta Committee:

"Many in the committee, myself included, have to double or triple up on duties. John has given so much time between the magazine, treasurer and membership duties. Indeed, had the committee not twisted his arm and co-opted him back to the committee for one more year while we try to find a treasurer, Quanta would have been in difficulty now and face the very real possibility of winding up the organisation.

It would be great to hear from people explaining why they DON'T want to go on committee. Surely it can't all be down to lack of time? We don't have to travel much to committee meetings, modern technology means we can do most work by email or telephone or conferencing and just meet up once or twice a year as necessary to sort out the major issues."

He then wrote about magazine copy problems:

"Quanta and QL Today have a "hard core" of contributors. Neither has paid contributors. With the possible exception of Rich and Jochen, nobody is making money from the QL, so it relies on volunteer effort. The few volunteers who are active are doing a good job (writers like George Gwilt, Norman Dunbar, Steve Poole and so on) writing articles but you can't rely on that small number of people.

For some time now I have been trying to write an email program and a paint program for QL systems. Time is the issue - I have so many commitments I just can't get the time in long enough bursts to be able to write. It's many weeks since I last had the chance to look at the editor part of the email program, I doubt I'll ever finish it at this rate. The graphics program is stalled by the lack of screen dump software for GD2 screens, a long time bugbear of mine.

Even if readers have negative contributions to make (by that I mean criticising the lack of innovation, stagnancy and so on) to magazines, it's still worth stoking the flames every now and then and writing something just to let people know there are QLers out there, not just the same old names filling the mags all the time."

He also wrote of the contacts he makes via his website:

"I get a lot of contact from "retro" users via my website.

The other type of contact I get is ex-QL users dusting off their QLs once more and realising that emulators etc open up a new world of nostalgia for them. I do try to steer them in the direction of QL Today and Quanta, but never really know how successful I am about pointing people in various directions. My pages about "returning to the QL", "PC/QL file transfer", "zip/unzip" and "serial links" get respectable numbers of hits, so there's obviously people looking at all this info, but I just don't know where it all leads in the end."

Rich Mellor also responded to many criticisms of the QL that had been made during the discussion:

"Well, contrary to popular opinion on the list, I still see a future in the QL market. Yes, many of

my 600+ QL customers have come to me to purchase a membrane, or items of software to collect, but many of them (50-60%) have gone on to purchase other items too, including games, disk drives and other items which suggest that they are doing more than just collecting computers.

It does not help that Quanta do not actively work to promote the QL apart from running the odd show (well, not to my knowledge). I mentioned some months ago about going to the Vintage Computer Festival, but it was down to me to suggest that they let me have some membership forms, and a banner to display, along with Quanta magazines - even then, I had to print off the membership forms to take with me.

The QL Wiki (and Norman's own QDOS Wiki) are probably some of the best marketing tools out there for promoting the QL and informing people what it is about - however, I get no input from Quanta and requests for financial assistance to help preserve QL software and make it available into the future to support existing users fell on deaf ears.

That said, it is not just Quanta. SellMyRetro offers a cheap way of selling second hand items, and traders and user groups can open a free store, and use it to advertise themselves. Unsurprisingly no-one has yet bothered to do this - all it takes is 10 minutes to register with the site, open a webstore, decide which category you want to be linked to (eg. Retro Computers -> Sinclair -> Sinclair QL), upload a banner and put some introductory text about what you do and a link back to your own site if you like.

The number of registered users on the site continues to grow daily and there are quite a few items being sold through the site now - it also gets fairly high results in Google, when searching for purchase sinclair ql (or similar), so is ideal to help promote the QL.

I guess what I am saying is that if you want the QL to have a future, then come on pull your fingers out and contribute to forums and sites outside of this list. I know that some people put a lot of effort and contribute in other ways (writing free software, articles and publishing magazines), but we all need to use the internet to promote it more widely and be willing to stick our necks out."

By this point the discussion had been going on for some time and as is usual in online debates ill discipline sets it. The thread became increasingly off topic with lengthy detours into PCs, Linux and cloud computing. What remained of QL discus-

sion mainly centred on software and a (portable) emulator and not always so practical.

Perhaps the mood could best be summed up by this contribution from **Malcolm Cadman**:

"The QL market will continue to be smaller than the Spectrum, Commodore or others; as the QL sold less well when it was first introduced.

Therefore the "customer base" is smaller.

Yet, potential users will be tempted back in, or new ones introduced by an easy way in.

This would need to be through an Emulator - either completely free and workable - or an Emulator on a time period trial, say of 60 days.

Working on the "host" of PC hardware.

Qemulator and QPC2 would be candidates for this, if the authors agreed.

Enhanced versions, to move on to, could always also be an option to get some income back for the authors.

The marketing should be fun and interesting, and yet quite aggressive - to appeal to people to try it out.

There is an audience for retro computers - of which the QL is one.

A software package, or easy access to free or low cost software, would be essential to tempt users in. As they need to see something to be done with the system - and will not be interested in finding it all out for themselves.

Once interest is kindled, then users will get interested in acquiring original hardware and original add-ons, as well as using the emulator(s) on PC hardware (or others).

Most users, on this list, are not a part of this target market - as most already have several QL hardware items, and a lot of software; plus long experience.

The latter need tempting in a different way, by some new hardware or software, for the 21st Century - as the QL was a 20th Century invention."

SOFTWARE

Roy Wood triggered the software discussion:

"The QL was a business computer in concept. Very few games of any note. Right now, as was pointed out earlier, there is no decent word processor and no other, modern, usable, software. QPC2 runs just fine on my W7 machine, but why would I use it? If I was a new user what would make me want to buy it and run it?

On my W7 and XP machines I run music software, word processing and spreadsheets, email, photo manipulation and website creation software. What is there on the QL side of things that

matches this? The QL was fun to program in SuperBASIC (I never got any further) and, back 18 years ago, would multitask when most PCs could not (nor can the iPad), but it was rapidly overtaken by modern hardware and software.

I will never abandon it completely because it taught me a lot of the fundamental principles of computing. I have a lot of affection for it and for many of the people in the QL community because we did all those shows and we went through all that but you have to take off the rose tinted goggles and take a hard look at what you are promoting - and then decide*

George Gwilt replied by describing his own use of the QL:

"I use QPC2 for doing horrible things like tax returns for which I wrote Archive programs a very long time ago. Rewriting for a PC or whatever is just not an option for me.

I also use QPC2 for programming, both in SBASIC especially for quick one off results, and in Assembler. I have tried Visual Basic on a PC and I did not like it. Assembler on Intel chips is pretty ghastly. A new user of QPC2, say, would, I imagine, almost certainly want to use it for programming - and almost certainly not for the word processors etc available."

John in Wales added:

"I entirely agree. The easy access to S*basic is one of the major trump-cards in the hands of QLers.

"Many of us began in a 'home-programming' type environment. Remember the thrill?

Taking on board the comments of Malcolm and other contributors I continue to believe (as I wrote in QUANTA some ? years ago) that to get the best in the QL community, novices or returners must be comfortable with QL community jargon."

Norman Dunbar partially agreed with Roy:

"The QL is dead, long ago. It was dead really before it started out in life in my opinion, but that never stopped me having 5 of the damned things!

It was a "32 bit" machine, erm no it bloody wasn't! It was a business machine - afraid not. The QL never knew what it actually was - unlike the Spectrum and ZX-81 (my other Sinclairs) which did have their own niche in the market.

There is no point, really, in bringing it back to life - it never had one to start with!

I think QUANTA was well named, Tinkerers, that's what we QL users are really. We tinker with a machine, making it do things it was never designed to do, and we have fun!

If we try to raise the QL above the hobby and

tinkerer level, we won't attract new users (in my opinion) simply because, as you say, you can do word processing, spreadsheets, graphics, photos, databases, video editing, programming in dog knows how many different languages, just about anything in fact. (Ok, maybe not magazine production, quality music or real graphic artistry - you'd most likely use a Mac for that) but the quality is far superior to anything the QL can provide.

But then again, the QL is circa 1984 and this is 2010 and Moore's Law still applies. Mind you, I'd love to see a PC doing so well with only 32KB or ROM and 128 KB of RAM!

The QL had one of the best Basic's around and still has in my opinion. In fact, SuperBasic is probably one of the best languages around, never mind Basic."

Urs König gave a mixed opinion:

"I will try to explain to my son the basics of a computer program by using SuperBASIC with QemuLator in fullscreen mode on our family notebook. Some FOR/NEXT loops (e.g. FOR i=1 to 7 step 2), some PAPER/ INK/ PRINT/ LINE/ CIRCLE calls using some variables, even some RND and BEEPing. But I'm sure that if he gets interested in programming we will move to a recent development environment very soon (in months if not weeks). He will definitely ask me: "How can I program my first iPod app?"

I like SuperBASIC very much and it was the programming language my career started with, but I must say that there's no need for S*BASIC in 20xx. Even with SMSQ/E think of its limitations like "stick with line numbers", very limited datatypes (only the string datatype is somehow 21st century, no 32bit integers, no usable floating point format), no modularity (you have to handle "modules" on your), no object orientation (OOP) at all, no IDE (ED is all we have built in, no debugger, nothing; OK, you can add QREF, other Toolkits, use QMON for S*BASIC trace/debug -> arghh, etc. pp.), GUI programming only as an add-on (QPTR, EasyPTR, TurboPTR).

In my opinion there are two drawbacks of modern platforms:

1. (Relatively) huge packages and therefore not that easy to get an overview.
2. BOOT up times like we love from the good old Sinclair computers are not possible."

Bob Spelton praised low level programming:

"It was seeing Basic programming demonstrated that sparked my interest in computers. I also read a few dozen of the many comments. Although some admit to starting off in ZX-

Basic, most seem to agree that Basic is bad for the brain and makes it more difficult to get your head around higher level languages like C++. These would allow thinking more like a human than like the machine. But it seems to me that they rely a lot on libraries of routines that do the actual work and somebody has to understand and speak at the machine level to make this happen.

Personally I think highly of those low-level programmers."

Stephen Usher also mentioned the value of low level programming:

"Line numbers, although a pain for advanced programmer, do help novices think about order. Data types confuse things and make it more complex than it needs to be.

In time, once the novice has grown out of the language they can move on. It's not as if we're saying to banish them."

And complex languages have disadvantages:

"Actually, the biggest drawback is complexity. It often takes a number of weeks learning the language and especially the overly complex library calls and hundreds of lines of code just to print "Hello world!" This is a major turn off for the teenage absolute novice.

If you can't look at a manual and get fun things happening within 5 minutes you've lost the battle and the war."

PIPE DREAM?

Dilwyn Jones started a new thread:

"I think it'd be great to have an "online QL" running in a browser - perhaps Java based or whatever. I seem to remember someone mentioning a ZX81 or Spectrum which would run in a browser.

That way, you'd be free of the nuances of any particular QL emulator or QL compatible - wherever you are, fire up your browser and access the "QL" over an internet connection. Synchronised online storage space (there's plenty of free space providers) would ensure your files would be up to date no matter whether you were running it at home or away from home."

Rich Mellor backed this up:

"I would also like a java based Sinclair QL emulator and perhaps that would be a project which Quanta could help fund the development of - it would attract a much wider audience and enable demos of programs to be played online to show what the QL is capable of.

There are already Java based emulators for the Amiga - perhaps someone could use this core,

or even see if they can get the QL emulator to run on the Java based Amiga emulator."

Darren Branagh was also enthusiastic:

"I agree with Dilwyn - an online browser based QL is very much needed - I use a ZX Spectrum one all the time via a Facebook app, great stress relief. :-) most of the games on World of Spectrum are available this way too.

I would love to see Marcel work on converting QPC to run this way - Marcel, is this possible? If so, is much work involved? I certainly wouldn't mind paying a few bob for the ability to pull up a working QL on ANY PC I happen to be working on - with an internet connection, or course."

Marcel brought the discussion down to reality:

"Well, if "a few" means about 50000 and "bob" is "Euros" I might start considering it, but otherwise, life's too short, sorry ;) I'm no student anymore, time is very precious these days. It would have to be written from scratch in any case. The only advantage I'd have is a somewhat intimate knowledge of the system."

But then as Dilwyn had written in his email:

"Ah well, I can but dream ..."

FOOTNOTE

All quotations have been subedited to correct spelling, grammar and other errors. To simplify following the various threads some contributions have been shortened and others are not in strict chronological order.

To subscribe to the QL-users email list you should send an email to:

ql-users-request@lists.q-v-d.com

with the message (not the subject) "subscribe".

And some final comments from Jochen:

I am glad to read that Rich Mellor is making money from the QL - so was I, but that stopped years ago, unfortunately. Looking at last year, the "income" from the QL would not even cover the costs for the payment infrastructure. Just being able to accept credit cards costs me about 700 EUR per year (base fee, terminal rent etc)... the year before, it was "only" 360 EUR (no transaction included!). They doubled the costs by forcing everybody to pay for Mastercard 3Dsecure and Verified-by-Visa, even if one does not use it.

Add it to all the other "basics" (website, telephone, fax, office and so on) I would not be able to survive one month from QL sales.

But - as you can see, I am happy to spend a lot my time producing QL Today!

Easy PEasy - Part 2

by Norman Dunbar

1. Introduction

At the end of the chapter 25 – Easy PEasy Part 1, I promised to take a look at the various code routines that George has written to make life a lot easier for PE assembly language programmers. If you haven't already done so, get over to George's web site and download the programs mentioned last time. The website address is

<http://web.ukonline.co.uk/george.gwilt>

2. Easy PEasy

As I mentioned last time, Easy PEasy isn't a program you can run, it is a collection of information and small binary files that you can include with your own programs – using the LIB and IN commands in your source code and assembling with GWASL – to make programming the Pointer Environment a little easier.

3. Supplied Files

With Easy PEasy, there are a number of files supplied, these are:

File	Description
Keys_pe	A file that can be included in your source file to define a number of equates for the various Trap #3 routines introduced by the PE.
Keys_wdef	Another include file. This one defines the WMAN window definition equates.
Keys_wman	Similar to keys_pe above but this file defines the equates for WMAN routines and vectors.
Keys_wstatus	This file defines the equates etc for the window status area.
Keys_wwork	This file contains the definitions for the window working definition.
Qdos_pt	The equates etc for the PE interface.
Csprc_bin	Some sprites, mostly for mode 4 but a few exist for mode 8. This file should be LIBbed by your own programs to use the sprites.
Csprc_sym_lst	This file lists the names of all the sprites in the above file. If you need to use a sprite in the above (binary) file, you must use the name listed in this file.
Peas_bin	This file contains all the useful code subroutines that George has written to make using the PE from assembly language easy. This file is binary and as such, should be LIBbed by your source code.
Peas_sym_lst	This file lists all the routines supplied in the above file. Make sure that you use the name(s) listed in this file if you wish to use George's code in your own PE programs.

4. Subroutines in Easy PEasy

The file peas_bin should be included at the very end of own program's code, as follows:

```
in      win1_source_easypeasy_peas_sym_lst
lib     win1_source_easypeasy_peas_bin
```

The first 'in' line includes the peas_sym_lst file which defines offsets from the current position to the entry points for the routines in the peas_bin file which is copied 'as is' straight into your final executable file. For this reason, you must keep these lines together and in the order shown above.

Routine	Description
GetSp	Allocates an area of memory and returns the address in A0.L. The size of the area required must be passed in D1.L on entry. No other registers are affected. Exits Via SUI (see below) if the memory allocation causes an error.
Rechp	Deallocates and frees an area of memory allocated by GetSp above. The address should be passed in A4.L. No other registers are affected.
Move	Processes a MOVE request then returns with D4 and D0 both set to zero. No other registers are affected. Can be called from inside the MOVE action routine in your own programs.
Sleep	Puts the program to sleep and creates a button in the button frame – if present. If the button frame is not present, the button will be placed on the top left of the display. See below for register usage.
Set_AP	Set an application window menu. See below for register usage. All registers are preserved on exit.
Sui	The program exits without warning and without any error messages. GetSp above will exit through here if there is an error when allocating memory.

4.1 GetSp

GetSp allocates an area of memory for the current job, and returns the address in register A0.L. There are no errors returned (in D0) as the routine exits through sui (below) if it detects an error. Only register A0.L is affected by the routine – all others are preserved.

On entry, the number of bytes required should be held in D1.L. On exit, A0.L holds the address of the allocated area. An example of use, taken from George's example EX0_asm, is:

```

...
move.l    #ww0_0,d1          Size of working definition.
bsr      getsp              Return ALCHP'd address in A0.
movea.l  a0,a4              Copy to A4.
...

```

There is no requirement to check for an error with this routine, if it returns to your program then it has worked.

4.2 Rechp

Rechp returns an area of memory, probably allocated using GetSp above, to the system. The address to deallocate must be passed in A4.L. All other registers are preserved and no errors are returned by this routine. An example of use would be after unsetting a widow definition, as per the following from EX0_asm:

```

...
jsr      wm_unset(a2)
bsr      rechp
...

```

Again, there is no need to check for errors as the routine never fails.

4.3 Move

Move is called when a program detects that the user has requested a MOVE be carried out. The routine can be called either from your own code (if the read pointer loop exits with D0/D4 not zero) or from within an action routine called by the read pointer loop. In either case, calling the move routine is as simple as this:

```

; MOVE loose item action routine.
afun0_0 bsr      move          Process a MOVE.
...

```


The above is another example taken from George's EX0_asm example program. After processing the move, the program needs to reset the loose item that caused the move request. See below for a fuller explanation of the example program and the code that is used to reset the loose items.

4.4 Sleep

Sleep sets the program to a button which contains the name of the program and is placed in the button frame if there is one or at the top left of the screen if there isn't.

While in button mode, A HIT – left mouse click or SPACE – on the button will cause the program to waken and restore itself to full size again.

A DO – right mouse click or ENTER – on the button will cause the program to waken if the program is currently located in the button frame, or, causes a move if the button frame is not present.

The registers required to call sleep are:

D1.L = The size needed for the button. (Can be obtained from ww0_1.)
 D2.L = The size needed for main window. (Can be obtained from ww0_0.)
 A2.L = The WMAN vector.
 A4.L = Pointer to the window working definition for the button window.

On exit from the sleep routine, the registers are set as follows:

D1-D3 = Undefined.
 A0.L = The channel ID
 A1.L = Undefined.
 A2.L = Preserved – the WMAN vector.
 A3.L = The window definition address.
 A4.L = Pointer to the working definition which may have changed.

As before, the following is an example from EX0_asm where the SLEEP loose item sets the sleep event in D4 and returns. This causes the read pointer loop to exit back to the user's code where the events etc are checked. The following extract shows the checks made to handle the sleep event being detected:

```
no_er2    ...
          btst      #pt_zzzz,wsp_weve(a1)  Was it a SLEEP event?
          beq.s     wrpt                    No, read the pointer again.
          move.l    #ww0_1,d1              Get main window button size.
          move.l    #ww0_0,d2              Get main window size.
          bsr       sleep                  Process a SLEEP.
          bra.s     wrpt                    Read the pointer again.
          ...
```

In the above extract, we can see D1 and D2 being set to the sizes calculated (by SETW – see previous chapter) for the main window and the buttonised window. Registers A2 and A4 are correctly set. After calling sleep, the program must continue to read the pointer otherwise it won't know if a DO or a HIT has been detected, or if it has been woken from slumber etc.

4.5 Set_AP

Set_AP is used to create an application window menu within a particular application window for a program. It is assumed that each item in the menu will be exactly the same length, although if QDOS strings are being used the word count for each one will determine what appears.

The registers required to call Set_AP are:

D1.W = How many items are present?
 D2.W = The length of each item.
 A0.L = Pointer to the start of the list of items.
 A1.L = Pointer to the application window.
 A4.L = Pointer to the window working definition.

On exit, all registers are preserved.

George has provided an example program that uses this routine, `EX1_asm`. When run, the program displays a list of files on `flp1_` when you click on the Display loose item. You can then select as many files as you wish, and click the Copy loose item. The selected files will then be copied to `ram1_`. The appropriate extract from this demonstration program is:

```
...
movea.l   fnmes(a6),a0           Pointer to list of file names.
moveq     #36,d2                 Interval between entries.
movea.l   a1,a5                  Needed later on, saved.
movea.l   ww_pappl(a4),a1       List of application window pointers.
movea.l   (a1),a1                Get application window zero from list.
bsr      set_ap                  Set the application Window menu.
...
```

The program has previously read the directory of all files (but no directories etc) on `flp1_` into the area of memory addressed by `A0`. The data stored there (effectively) looks like the following:

```
item_1    ...
dc.w      4                       Length of string
dc.b      'boot'                  Filename from flp1_
dc.b      0,0, ...                32 padding bytes.

item_2    dc.w      7               Length of string
dc.b      'boot_pe'              Filename from flp1_
dc.b      0,0, ...                29 padding bytes.
...
```

You can see from the above that each entry is a total of 36 bytes long (the difference between addresses `item_1` and `item_2`) although the actual menu items themselves, the filename, need not be exactly 36 bytes. Regardless of the value of the padding bytes, the data displayed in the menu items will only show the actual filenames as defined by the QDOS strings making up each item.

An article on application window menus will be coming soon in this series.

4.6 Sui

`Sui` is a dramatic routine to call. Wherever your program is in its processing, calling `sui` will cause it to exit. In addition, the `GetSp` routine (above) will call `sui` if it cannot allocate a suitable area of memory. George's example program calls `sui` when it detects that the Pointer Environment is not present, as follows:

```
...
moveq     #iop_pinf,d0           Find Pointer Environment & WMAN
moveq     #-1,d3                 Timeout.
trap      #3                     Do it.
tst.l     d0                     Did it work?
bne       sui                    No failed, or PE absent, bale out.
...
```

No registers are used by this routine. It never returns an error – because it never actually returns!

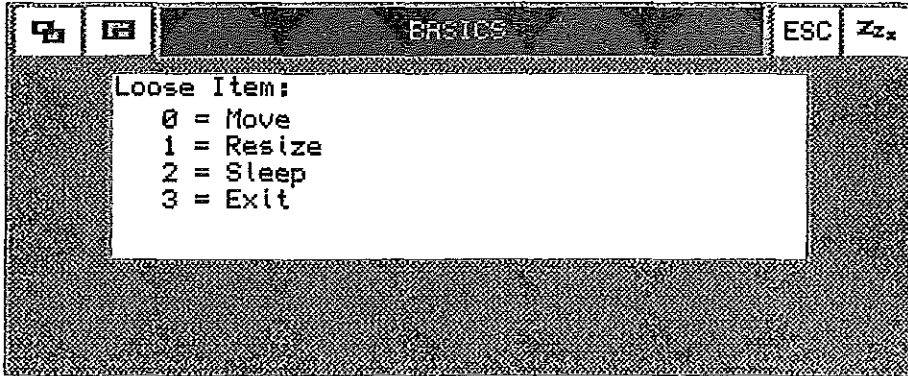
5. The Example Program, `EX0_asm`

So, having discussed the various bits and pieces of Easy PEasy, lets dissect one of George's example. The simplest example is `EX0_asm` and it's corresponding SETW designed window file, `EX0w_asm`, so those are what we will look at next.

This example simply shows how to use the four main events in a PE program:

- Move – moves the window around the screen.
- Resize – allows the window to be sized.
- Sleep – puts the program to sleep either in the button frame, if present, or on screen.
- Esc – exit from the program.

The program looks like this when running on QPC:



Example program EXO in action

The window above shows an outline with a green border and an interesting paper colour. A white information window is displayed listing the various loose items in the program. The information window is white with a green border and black ink.

Along the very top of the window we can see the program's title – BASICS – in a red papered information window with a green border and ink, and the four loose items for MOVE and SIZE on the left with ESC and SLEEP on the right.

The window has been created by SETW and the definitions are all held in the file `EXOW.asm` which is supplied in the peass download from George's web site.

What follows is a slightly amended version of the program supplied by George. I have updated some of the comments to make them more readable and understandable (by me!) and in a couple of places, I have rearranged the order of some of the instructions – with George's blessings of course.

```
;  
; Standard job header.  
;  
        bra.s    start  
        dc.l    0  
        dc.w    $4afb  
  
fname   dc.w    fname_e-fname-2  
        dc.b    "EXO v1.05"  
fname_e ds.b    0  
        ds.w    0  
  
;  
; Include the various Easy PEasy include files. These give us names for all  
; the various offsets, vectors, traps etc used by the PE.  
;  
        in     win1_ass_pe_keys_pe  
        in     win1_ass_pe_qdos_pt  
        in     win1_ass_pe_keys_wwork  
        in     win1_ass_pe_keys_wstatus  
        in     win1_ass_pe_keys_wman  
        in     win1_ass_pe_keys_wdef
```

```

;-----
; Define a few explicit equates for this example program. These are offsets
; into the program's dataspace (relative to A6) where we store various bits
; of useful information, channel ids and so on.
;-----
id      equ      0
wmvec   equ      4
slimit  equ      8                               Size - origin

```

The above is the usual QDOSMSQ job header, and so on. The various in lines pull in the include files from Easy PEasy.

```

;-----
; Here is when the example code really starts.
;-----
start   lea      (a6,a4.1),a6                    Dataspace in A6.
        bsr.s    ope                            Open a con channel.
        move.l  a0,id(a6)                       Keep the ID safe.
        moveq   #iop_pinf,d0                    Find Pointer Environment & WMAN
        moveq   #-1,d3                          Timeout.
        trap    #3                              Do it.
        tst.l   d0                              Did it work?
        bne     sui                             No failed, or PE absent, bale out.
        move.l  a1,wmvec(a6)                    Keep WMAN vector safe too.
        beq     sui                             WMAN not present, bale out.
        movea.l a1,a2                           Copy WMAN vector to A2.
        lea     slimit(a6),a1                   Buffer for results.
        moveq   #iop_flim,d0                    Find maximum size of window.
        trap    #3                              Do it.
        subi.l  #$C0008,(a1)                   Less 12, 8 from width, height.
        lea     wd0,a3                          Address of main window definition.
        move.l  #ww0_0,d1                       Size of working definition.
        bsr     getsp                            Return ALCHP'd address in A0.
        movea.l a0,a4                           Copy to A4.

```

The section of code above carries out various initialisations and checks for the Pointer Environment and WMAN before allocating enough space for the working definition for the main window which SETW stores for us in ww0_0.

```

;-----
; We need to set the status area to zeros
; and the loose items to "available" (zero)
;-----
        lea     wst0,a1                          Status area address.
        movea.l a1,a0                            Copy to A0.
        moveq   #wst0_e-wst0-1,d1               Bytes to clear - 1.

st1     clr.b   (a0)+                            Set status to zero/available.
        dbra   d1,st1                            And repeat.

        movea.l id(a6),a0                        Get the channel ID again.
        move.l  wd_xmin+wd_rbase(a3),d1         Minimum size (x,y) in D1 (hi,lo).
        andi.l  #$OFFFOFFF,d1                   Lop off the scaling factors.
;-----
; Wm_setup gets upset if you leave the
; scaling stuff attached. The x,y
; sizes in D1 must be actual sizes.
        jsr     wm_setup(a2)                     Set up the working definition.

```

Just before we (finally) set up the window, we need to be sure that all the loose items are set to available - in this case - and that the status area is filled with zeros. As ever, SETW has put the status

area details in an easy to find location – wst0 – and we use this to initialise the status area easily. Regardless of the actual size of the status area itself, the above code will always work.

Please note, in the above George picks the smallest window definition as the one to use when the program first starts. The size of the smallest definition is obtained from wd_xmin+wd_rbase(a3) and placed in D1.L with the high word containing the width and the low word holding the height. Because this definition has scaling details embedded in the top nibble of each word, these must be masked out before calling wm_setup.

The same applies if you set D1.L to zero – which means use the default (largest) definition – unless the scaling factors are masked off, the call to wm_setup will return, but your window will not display correctly, if at all. This problem also affects the wm_fsize routine which returns the size, in D1.L, for a given definition. You must mask off the scaling nibbles.

```

moveq    #-1,d1                Set the window position ...
jsr      wm_prpos(a2)          ... to where the pointer is.
jsr      wm_wdraw(a2)         Draw the contents.

```

The snippet of code above sets the window position to be where the pointer is on screen right now, then draws the window.

```

wrpt     jsr      wm_rptr(a2)    Read the pointer.

```

The above starts the pointer reading loop. This code will not return unless an action routine sets D0 with an error code, or, sets D4 with an event number.

```

beq.s    no_err                As D0 is zero, D4 must be non zero.
bra      sui                    Error, D0 is non zero, bale out.

```

If we have returned from the read pointer loop, then D0 is holding an error code, or D4 holds an event number. Because the Status Register must hold the flags according to the value in D0 on exit from an action routine, checking for the Z flag being set implies that D0 is indeed holding an error.

If no error is detected, the code skips off to a label no_err below, where D4 is checked for events to process, otherwise, the program dies horribly with a call to the sui routine supplied by George.

```

;-----
; Default console channel definition.
;-----
con      dc.w      3
         dc.b      'con'

;-----
; Routine to open a channel for this job.
;-----
ope      lea      con,a0        To open "con"
         moveq    #-1,d1        For this job
         moveq    #0,d3
         moveq    #io_open,d0
         trap     #2
         rts

```

The code above defines a console channel for our program and opens it.

```

no_err   movea.l   (a4),a1      Status area
         btst     #pt__can,wsp_weve(a1) Was it a CANCEL event?
         bne     sui            Yes, exit

         btst     #pt__move,wsp_weve(a1) Was it a MOVE event?
         beq.s   no_er1        No, skip.
         bsr     move        Yes, process a MOVE.
         bra.s   wrpt         Read pointer again.

no_er1   btst     #pt__wsiz,wsp_weve(a1) Was it a SIZE event?
         beq     no_er2        No, skip.

```


	bsr.s	resze	Yes, process a SIZE.
	bra.s	wrpt	Read pointer again.
no_er2	btst	#pt_zzzz,wsp_weve(a1)	Was it a SLEEP event?
	beq.s	wrpt	No, read the pointer again.
	move.l	#ww0_1,d1	Get main window button size.
	move.l	#ww0_0,d2	Get main window size.
	bsr	sleep	Process a SLEEP.
	bra.s	wrpt	Read pointer again.

The code above is executed on return from the read pointer loop with an event number in D4. It begins by checking to see if the CANCEL event occurred (or was set in an action routine) and if so, exits the program via the sui routine.

Assuming that the event was not CANCEL, the next check is for a MOVE event. If it was a MOVE, the move is handled by George's move routine and we return to the read pointer loop again.

The next check is for a SIZE event and if detected, we process the MOVE request and return to the pointer reading loop, otherwise we skip to the final check.

The last check we make is for a SLEEP event. If this is not a SLEEP request, we skip back and begin reading the pointer again. If this is a SLEEP request, we set the registers as required by the sleep routine by loading D1 with the current window size and D2 with the button window size – both helpfully defined by SETW – and jump into the sleep routine.

The sleep routine returns control to our code again and we skip back to reading the pointer. We must do this or we will never be able to know when the sleeping program has been wakened etc.

All of the above checks were made by looking at the individual bits in the window byte of the event vector.

```

;-----
; Loose item action routines.
;-----
; MOVE
;-----
afun0_0 bsr      move      Process a MOVE.

af1     move.w   ww1_item(a3),d1      Loose item number.
        move.b   #wsi_mkav,ws_litem(a1,d1.w) Ask for redraw.
        moveq   #-1,d3              Selective redraw.
        jsr     wm_ldraw(a2)         Redraw loose items.
        clr.b   ws_litem(a1,d1.w)   Available status.
        moveq   #0,d4              No events.
        moveq   #0,d0              No errors.
        rts     Go back to reading the pointer again.

```

The program demonstrates both methods of handling loose item action routines. MOVE and SIZE are handled within the read pointer loop and not by the above code which checks the event bits outside of the read pointer loop.

The action routine above, for a MOVE, carries out all the processing necessary to make the window move on screen. It simply calls the move routine supplied by George.

The code at label af1 is necessary as it resets the loose item's status to available – when a loose item is hit or done, it's status changes to selected. Once this has been done and the loose item redrawn, D4 and D0 are set to tell the read pointer loop to continue, the action has been processed.

```

;-----
; RESIZE
;-----
afun0_1 move.l   a3,-(a7)           Save working register.
        movea.l  ww_wdef(a4),a3    Window definition x,y size.
        bsr.s   resze             Process a SIZE.
        movea.l  (a7)+,a3         Restore pointer to loose item.
        bra.s   af1              And reset status etc.

```

The action routine above processes a SIZE request when the Size Loose Item is hit or done. It does this by calling code common to the action routine itself and called by the user level code (outside the pointer reading loop) when a SIZE event bit is set in the window event vector.

Unfortunately, there is no Easy PEasy way to do a resize (at least, not at the moment) so we programmers have to do it all ourselves. As shown below.

```
;  
* To perform the resize we need to  
* a. Find the amount of resize (by wm_chwin)  
* b. Throw away the current working definition (by wm_unset)  
* c. Find the new size (by wm_fsize)  
* d. Get space for the new working definition (by getsp)  
* e. Set up the new working definition (by wm_setup)  
* f. Position the new window (by wm_prpos)  
* g. Draw the contents (by wm_wdraw)  
*  
* Comments  
* On a.  
* We have to set the resize bit in the window byte of the  
* event vector in the status area before wm_chwin is called.  
* The change in size is returned in D1 and the window size  
* event number is returned in D4.  
*  
* On c.  
* On entry to wm_fsize, D1 must contain the requested size.  
* This size must be chosen carefully. It must be no bigger  
* than the maximum in the window definition. It must be  
* smaller than the maximum size for the window layout. The  
* x-size must be a multiple of 4 (to allow proper stippling.  
* Finally the size must not be bigger than the current screen  
* size with allowance for shadow and border.  
* On exit D1 contains the actual size and D2.W contains the  
* number of the repeated section.  
* In this example we do not really need to use wm_fsize since  
* we know that D2.W will be zero and that the value in D1  
* will be that on entry (since we have a variable window).  
*  
* On d.  
* The space needed is found from the label ww0_0 set in the  
* window definition.  
*  
* On e.  
* For wm_setup we need on entry:  
* D1 = size  
* A0 = channel ID  
* A1 -> status area  
* A3 -> window definition  
* A4 -> space for working definition  
*  
* On f.  
* On entry to wm_prpos we need the position in D1.  
* In order to ensure that the bottom right corner of the  
* resized window is in the same position as that of the old  
* we need to subtract the increase in size from the pointer  
* origin in the old window.  
* The new position is thus wd_org plus ww_xsize minus the  
* new size.  
*  
;  
resize    move.l    ww_xorg(a4),d7  
          move.l    ww_wdef(a4),a5                Window def
```

```

add.l    wd_xorg(a5),d7
add.l    ww_xsize(a4),d7      Ptr position for PRPOS (optr)
bset     #pt_wsiz,wsp_weve(a1)
jsr      wm_chwin(a2)        Sets change to D1 (mv)
bclr     #pt_wsiz,wsp_weve(a1)
move.w   wd_rbase+wd_xmin(a5),d5
andi.w   #$fff,d5
move.w   ww_xsize(a4),d4
swap     d1
sub.w    d1,d4
cmp.w    d5,d4
bgt.s    resze1              D4 is greater
move.w   d5,d4

resze1   move.w   wd_xsize(a5),d5
         cmp.w    d5,d4
         blt.s    resze2              D4 is smaller
         move.w   d5,d4

resze2   moveq    #3,d3
         add.w    d4,d3
         andi.w   #$fffc,d3          Keep answer in D3.W (mv)
         move.w   wd_rbase+wd_ymin(a5),d5
         andi.w   #$fff,d5
         move.w   ww_ysize(a4),d4
         swap     d1
         sub.w    d1,d4
         cmp.w    d5,d4
         bgt.s    resze3              D4 is greater
         move.w   d5,d4

resze3   move.w   wd_ysize(a5),d5
         cmp.w    d5,d4
         blt.s    resze4              D4 is smaller
         move.w   d5,d4

resze4   swap     d3
         move.w   d4,d3              D3 = new mv x | y
         jsr      wm_unset(a2)
         bsr      rechp

```

```

; Now restrict size to the screen size less (12,8)
;

```

```

         move.l   slimit(a6),d1
         cmp.w    d3,d1
         ble     resze7              D1 OK
         move.w   d3,d1

resze7   swap     d1
         swap     d3
         cmp.w    d3,d1
         ble     Resze8              D1 OK
         move.w   d3,d1              New size

resze8   swap     d1                  New limited size
         move.l   d1,d3
         jsr      wm_fsize(a2)
         move.l   d1,-(a7)           Keep size pro tem
         move.l   #ww0_0,d1         Space needed
         bsr      getsp

```

```

move.l    (a7)+,d1           Replace size
movea.l   a0,a4             New wwd
movea.l   id(a6),a0        Replace ID
jsr       wm_setup(a2)

```

```

;-----
; The position for PRPOS is optr-mv with minimum of 4 | 2
;-----

```

```

move.l    d7,d1
swap      d1
swap      d3
sub.w     d3,d1
cmpi.w    #4,d1
bge.s     resze5           D1 not less than 4
move.w    #4,d1           Set minimum of 4

```

```

resze5    swap      d1
swap      d3
sub.w     d3,d1
cmpi.w    #2,d1
bge.s     resze6           D1 not less than 2
move.w    #2,d1           Set minimum of 2

```

```

resze6    jsr       wm_prpos(a2)
          jmp       wm_wdraw(a2)

```

The above code is George's way of processing a SIZE request from within an action routine or from user code that detected the SIZE bit set in the window event vector.

The other two action routines, for SLEEP and ESC, demonstrate how an action routine can simply set the appropriate bit in the window vector, set D4 to indicate an event and exit with D0 cleared.

In this case, the actions cause the pointer reading loop to return to the user's code where the events can be checked for (see above) and processed accordingly.

```

;-----
; EXIT - set the CANCEL event in the windows event vector, put the CANCEL
; event number in D4 and exit with D0 set to zero.
;-----
afun0_3   bset      #pt__can,wsp_weve(a1)   Set CANCEL bit in the window event.
          moveq     #pt__can,d4            ESC event number.
          moveq     #0,d0                  No errors.
          rts                               Return to exit from reading the
;                                           pointer and into the PROCESS EVENT
;                                           section of the user's code.

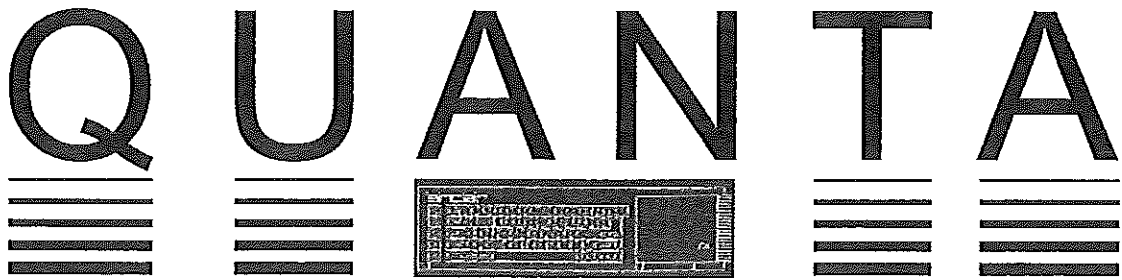
```

First of all, the action routine for the ESC loose item. This is the simplest action routine as it only has to set the event bit, set D4 and D0 then exit. It doesn't have to reset the ESC loose item status from selected back to available because the program is about to exit and the user will never see the redrawn loose item. Simple.

```

;-----
; SLEEP - set the ZZZ event bit in the window event vector, put the ZZZ
; event number in D4, redraw the ZZZ loose item as available - otherwise it
; is still selected when we wake from the button frame - then exit with D0
; set to zero.
;-----
afun0_2   move.w    ww1_item(a3),d1        Item number for the ZZZ loose item.
          move.b    #wsi_mkav,ws_litem(a1,d1.w) Ask for redraw to available.
          moveq     #-1,d3                  Selective redraw.
          jsr       wm_ldraw(a2)           Redraw loose items.
          clr.b     ws_litem(a1,d1.w)      Available status set.

```



Independent QL Users Group

World-wide Membership is by subscription only,

Offering the following benefits:

Bimonthly Magazine - up to 52 pages

Massive Software Library - All Free!

Free Helpline and Workshops

Regional Sub-Groups. One near you?

Advice on Software and Hardware problems

Subscription just £14 for Full Membership

PayPal (see QUANTA Web Site),
Cash, Cheques and Postal Orders Accepted

Now in our Twenty Seventh Year

Further details from the Membership Secretary

**John Gilpin, 181, Urmston Lane,
Stretford, Manchester, M32 9EH (UK).**

Tel. +44 (0) 161 865 2872

Email: membership@quanta.org.uk

Visit the QUANTA Web Site

<http://www.quanta.org.uk>


```

    bset      #pt_zzzz,wsp_weve(a1)  Set ZZZ bit in the window event.
    moveq    #pt_zzzz,d4             ZZZ event number.
    moveq    #0,d0                   No errors.
    rts                                     Return to exit from reading the
;                                     pointer and into the PROCESS EVENT
;                                     section of the user's code.

```

The sleep loose item's action routine is almost as simple, but because the program will – hopefully – be awakened at some point, it has to reset the loose item status and redraw it.

The code above starts off by obtaining the correct loose item number and changing it's status to indicate that it is available. It then calls `wm_ldraw` to redraw only those loose items asking for a status change & redraw – as signalled by the value of minus one in D3. This prevents redrawing up to 32 loose items which don't need redrawing because nothing has changed.

Once redrawn, the loose item's status is set to available as well, the SLEEP bit is set in the window event vector, D4 is set to show the event number and we exit with D0 cleared to show that no errors occurred.

On return from the above two action routines, the read pointer loop will exit and processing will continue from the 'beq.s no_err' just after the `wrpt` label. (Many lines above!)

```

;-----
; Pull in window definition as created by SETW.
;-----
    in      win1_ass_pe_EX0w_asm

;-----
; Pull in the Easy PEasy stuff next – code routines and sprites.
;-----
    in      win1_ass_pe_peas_sym_lst
    lib     win1_ass_pe_peas_bin

    in      win1_ass_pe_csprc_sym_lst
    lib     win1_ass_pe_csprc_bin

```

The last few lines of code pull in the SETW defined window from the file `EX0w_asm`, then LIBs in the Easy PEasy routines and the various sprites that your program might want to use.

6. Coming Soon

As promised above, the next chapter in the ongoing saga of writing PE programs in assembler, will concentrate on application windows and application window menus.

Artificial Intelligence

by Stephen Poole

Editor: It seems we have made a mess of Steve's articles - this one should have been printed before his article in Issue 4 of QL Today Volume 14. Apologies to Steve and to the readers. Another article is also waiting. Looking at the current page number (40), and the fact that QL Today issues were planned to have an average of about 32 pages, and the fact that I need to insert more pages to accomodate Steve's current article, it is clear that the other article needs to wait for the next issue. Sorry about that too.

In America, there are two views of how the Universe originated. The oldest is the biblical description in the book of genesis, called 'creationism'. The second is the scientific description of evolution, which applies not only to living things, but also to all forces and matter.

Creationists consider that life is so complex that it must have been created perfectly by an intelligent God, that is, using 'intelligent design'. Science has analysed most aspects of reality in far greater detail, and has come to the conclusion that the universe is so old that the mere

forces of nature combined with probability have had the time to generate ever more complex structures into what we see today. Take the case of man : For creationists man is perfect, being in 'God's image', but for Science man is imperfect, as continuous and numerous birth defects prove, but on average, evolution has given man enough variability to survive from one generation to the next, imperfect as he may be.

In a military sense, intelligence is just masses of raw data collected far and wide. For the ancient Greek, Asiatic and Jewish philosophers, intelligence was the quest for Wisdom, distilled from generations of study and debate of all gathered knowledge, honed into a set of recommendations of how to live life in a harmonious way. Religious intelligence is none other than a vast knowledge of all the world's sacred texts.

Modern psychologists test people's brains to evaluate their IQs, (intelligence quotas), which allows them to measure their ability to come to logical deductions, which is probably most people's idea of what intelligence means. IQ is not a measure of Wisdom or general knowledge, just a person's potential to become intelligent as he grows older.

We could study the 'clever' way stars produce all known elements from hydrogen, but this would be to adopt creationists notions of intelligent design, (which leaves room for little comment). Instead, we can look at the more productive attitude of Science, which examines which individual natural laws canalise matter and energy within set semi-random bounds to produce organised organic matter: in particular the DNA helix, the matrix of life. Recently, genetics has proved the veracity of evolutionary theory beyond all doubt, by comparing the genes of many species.

DNA can be thought of as intelligent, as it sorts through the genetic code in a cell, choosing the best genes and repairing damaged ones. How it does this is not yet fully understood, as the interactions between the main DNA code and its associated RNA sequences have not yet been fully investigated in view of their complexity. Viruses are the simplest life-forms, but as they are parasitic we shall not consider them here, as they are incapable of independent existence. So let's move on to bacteria: These life-forms, simple as they are, reveal an astonishing adaptation to most environmental conditions, from freezing temperatures to boiling ones, from light to totally dark situations, from acid to alkaline waters, and from rich organic food to metallic chemical 'food',

and from lightweight to intense pressure situations. From a Darwinian standpoint, adaptability is the main feature of intelligence, but can we say that bacteria think? If we observe the way in which they move towards food sources, avoid dangers and use various collective strategies to survive we might well assume that they are indeed astute. On the next evolutionary level we may consider polycellular animals such as sponges or plants. These respond very actively to different essential stimuli and can communicate with one another using external chemical signals to indicate food sources or predators, much in the same way that internal human organs communicate using hormones. They have no nerves so no brains, but they do have collective organisation to a great extent. One step further up the evolutionary ladder lie vertebrates with a spinal chord, nervous system and a rudimentary brain, such as fish. Anyone who has kept fish in an aquarium can testify to the fact that goldfish can be trained to respond to signals from humans and that they have a considerable range of complex behaviour patterns. Unfortunately, we generally think of fish as dull creatures that get easily hooked by lures or as huge shoals that cannot avoid being trapped in nets. (In general we have a pretty poor impression of the animals we eat). The fact is that in nature there are no nets nor hooks, so fish have no fear of them. But in situations familiar to them they can reveal a great deal of cunning and behavioural adaptation.

Then on to birds. Many parrots have been taught to 'speak', as everybody knows, 'Parrot-Fashion'. But this is because man has only recently learned how to teach birds to answer logically. Crows are excellent problem-solvers that can use all manner of tools to obtain food. They inspect a problem such as a chinese puzzle containing a peanut, turn their heads as they ponder, then work out a strategy to get at the food by solving the puzzle! And all this without language..? Yet recently people have taught African Grey Parrots to say several hundred words, to combine them using grammar and reply logically after reflection, and even to do simple arithmetic... This came as a great shock to many people, as language is supposed to be the thing which separates man from animals, yet here was proof that animals with minute brains were capable of thought and reflection and even lie-telling!

So on and up another step to Washoe, the bonobo ape who could use a specially devised computer keyboard to interact with her keepers. She even taught her offspring to do the same applying many hundreds of words! Now apes, like

wolves are capable of great feats of collective tactics and strategy to obtain their goals, and live in complex societies, like early mankind.

Man was hardly different to other apes throughout great swathes of geological time until he perfected language enabling him to pass on detailed information from person to person in the same, if more complex way that bees or ants do. This enabled him to break the rigid social hierarchy imposed by the dominant males and allowed him to begin to form complex social groups who could make better use of all available resources and construct an important cultural tradition based on exchange. This was vital in hunting and food-gathering groups who were inter-dependant for tool-making and sharing resources such as flints or ochre trading for decoration.

When agriculture began to evolve from simple forest gardening to clearing, planting and harvesting fields of cereals, man began to store food, resources and knowledge enabling massive population growth. Civilisation and cities occurred, and remained largely based on the same tribal rules until the industrial revolution discovered how to harness fossil energy resources, when the world's population exploded liberating numerous scientists to develop knowledge and technology to its current exponential levels. (The downside of intelligence has been global warfare. Even warfare demands intelligence in the form of alliances, but real intelligence stops at such cooperation, thereby avoiding war altogether).

But fossil fuels and many other essential resources will definitely run out in the near future, creating massive pollution, economic and climate change on a scale unknown to man previously. We can use computer models to predict future trends, but what we need most urgently is the intelligence to face up to the facts, persuade everyone of the need to behave as responsible eco-citizens and make the right choices so as to prove that man is an intelligent species after all.

What does all this have to do with the QL and computing? Well, for the past forty years there has been sporadic talk of Artificial Intelligence, that is, teaching computers and robots to become the super-brains that will save us all from all of our problems. But how far has this got?

'Big Blue' managed to beat Kasparov in a chess tournament by a short head, but big blue used neither tactics nor strategy to win, just a huge database of all the winning moves during the last thirty years of the world chess series. Machine translation programs are still very poor when compared to human translators as they

have no background knowledge of context nor subtlety. Robots are only just beginning to acquire any real degree of autonomy, still requiring some remote-control, and compared to a new-born gnu they are hopelessly handicapped. Self-learning AI systems are mainly good at extracting trade secrets from unsuspecting experts in their respective domains to declassify them in their hierarchy.

So we can see that Artificial Intelligence has a long way to go before it can ape evolutionary intelligence, so we must still rely on good, sound human education to produce results and to ensure the future of our world's civilisation. In my article 'Machine Logic' I mentioned that the human brain contains some 50,000,000,000 neurons. One can add that the human brain in actual fact contains millions of billions of synapse junctions, meaning we are an incredibly long way from matching its performance. In the meantime, we can still study AI to help us, for example to improve internet user-friendliness, which could help old people to keep in touch with their relatives and improve their lives. The \$75 One Laptop per Child is now with us and will be on sale in the spring of 2009. And 860,000,000 starving people in this market-based world of ours could certainly benefit from a free internet laptop computer showing them how to escape from their chains of poverty.

Artificial Intelligence programs are designed to make computers think, that is, to make decisions by themselves based on data fed into them. One of the oldest examples of reasoning is the Syllogism of the ancient Greeks. Given two or more statements, one can deduce logical conclusions if both ideas contain common elements and therefore it is possible to make a coherent third statement. This is what this Think_bas program does.

To keep it simple, the routine only acts on a few statements. But see how by searching associations between sentences it can make a whole list of deductions. But its weakness is that it cannot act on hunches like the human brain, so poor old Plato and Diogenes are left out in the cold, as the QL doesn't guess that wise-men are men too...

Feel free to modify the code to add and test extra data such as 'A MAN IS A LAVATORY CLEANER' to see what the QL thinks about Socrates!. (Was Socrates class-conscious?). This will show you that 'artificial' Syllogisms only work correctly when you select your vocabulary very carefully.

I was careful to choose statements that are compatible, using but one verbal form. 'A man is

a mortal' is much easier to deal with than 'All men are mortals', which, if treated in the same way as with this program, would give the reversed deduction that 'All men are Socrates', clearly absurd. I also used DATA statements to avoid having to parse long sentences. This greatly simplifies the coding, as does replacing spaces with underscores making it clearer for you to understand, but beware of case errors! The crux of the routine is the cross-reference 'flagging' to indicate associations between phrases.

To really impress your friends, instead of just printing the deductions as we do here, get them to type in a question, for example, 'IS SOCRATES A PHILOSOPHER?'. The program should parse the question, inverse the subject and verb, look up the list of QL deductions in the array and prints its reply to the screen. Yes! the beast really is thinking, making deductions and storing its conclusions in memory just as we do...even though in a far less sophisticated way.

The human brain works mainly by the association of ideas, but the main weakness of AI systems is the same as that for translations: Computers do not yet have an adequate mastery of logic and grammar to be able to parse fully efficiently. There is plenty of scope for research

in this domain. Professional expert systems act on hundreds or thousands of statements, and they may also have to use multiple conditions before coming to correct decisions. Therefore they can come to conclusions rapidly, but can still only make the choices for which they are rigorously programmed. So their degree of intelligence is still far below that of the average man...who, (like our womenfolk), are capable of intuition.

It can be argued that the exact definition of intelligence must be taken from Darwin's theory of evolution: It will only be possible to say that man is intelligent after he has survived for hundreds of generations after the present potential environmental crisis. If we extrapolate this definition to computers, computers will be deemed intelligent when they are capable of self-sustained survival. But can you imagine a world where no electronic machinery can become obsolete? We would live in homes jammed full of out of date gadgets that refuse to die. The QI what!

Will Artificial Intelligence really help us to feed the World and avoid an ecological catastrophe? It could scarcely do worse than we are doing now..

```

100 ::
110 REMark Think_bas, by S,Poole, v8sept2008
120 REMark For QL Today. Beta-test by Bruno Coativy.
130 :
140 CLEAR: OPEN#1,con_16: WINDOW 512,256,0,0: PAPER 0: INK 7: CLS
150 RESTORE : lines=5: cols=3: words=8: chars=16
160 DIM p$(lines,cols,chars),t$(words,words,chars)
170 l$='': kt=1: w$='': left=1: Fc=9
180 :
190 REMark Increment 'lines' & 'words' if you add data:
200 DATA 'Socrates_', 'is_', 'a_man_'
210 DATA 'a_man_', 'is_', 'a_mortal_'
220 DATA 'a_mortal_', 'is_', 'a_philosopher_'
230 DATA 'Plato_', 'is_', 'a_wise_man_'
240 DATA 'Diogenes_', 'is_', 'a_wise_man_'
250 :
260 REMark Get the data:
270 FOR f=1 TO lines
280   FOR j=1 TO cols: READ p$(f,j): AT f,j*12: PRINT p$(f,j)!!
290   :
300   REMark concatenate the list of words:
310   FOR j=1,cols
320     Q$=p$(f,j): IF Q$ INSTR l$: ELSE l$=l$&Q$&'*': kt=kt+1
330   END FOR j
340 END FOR f: k1=kt-1: i$=INKEY$(#1,500): CLS
350 :
360 REMark Fill right and under array cells with words:
370 FOR f=1 TO k1
380   i='*' INSTR l$: right=i-1
390   FOR j=left TO right: w$=w$&l$(j)

```

```

400 l$(i)=' ': t$(f,words)=w$: t$(words,f)=w$: left=i+1: w$=''
410 :
420 REMark Print the grid:
430 FOR j=1 TO k1: AT j,f*Fc: PRINT '.'
440 AT f,72: PRINT t$(f,words)(1 TO 4)
450 AT words,f*Fc: c$=t$(f,words): PRINT c$(1 TO 8)
460 :
470 REMark Self-test:
480 FOR j=1 TO k1: IF f=j: t$(f,j)='0': AT j,f*Fc: PRINT '0'
490 END FOR f
500 :
510 REMark Make a cross-reference grid:
520 FOR f=1 TO lines
530 a$=p$(f,1): b$=p$(f,cols)
540 :
550 FOR ac=1 TO k1
560 FOR dn=1 TO k1
570 n1=0: n2=0
580 IF a$=t$(dn,words): n1=dn
590 IF b$=t$(words,ac): n2=ac
600 IF n1 AND n2: t$(ac,dn)='1': AT dn,ac*Fc: PRINT '1'
610 :IF n1 AND n2: t$(dn,ac)='1': AT ac,dn*Fc: PRINT '1'
620 END FOR dn
630 END FOR ac
640 END FOR f
650 :
660 REMark Link associated words:
670 FOR ac=1 TO k1
680 FOR dn=1 TO k1
690 IF t$(ac,dn)='1' THEN
700 REMark lookup next links:
710 word$=t$(words,ac)
720 FOR j=1 TO k1
730 REMark lookup referenced word on right:
740 IF word$=t$(j,words) THEN
750 REMark scan across for digit:
760 FOR f=1 TO k1
770 IF t$(f,j)='1' THEN
780 t$(f,dn)='1': AT dn,f*Fc: PRINT '2'
790 END IF
800 END FOR f
810 END IF
820 END FOR j
830 END IF
840 END FOR dn
850 END FOR ac: i$=INKEY$(#1,250)
860 :
870 CLS: PRINT 'Here are my deductions :'\
880 FOR f=1 TO k1
890 FOR j=1 TO k1
900 IF t$(f,j)='0' OR t$(f,j)='1' OR t$(f,j)='2' THEN
910 across$=t$(words,f): down$=t$(j,words)
915 ok=CODE(across$(1))<97 AND CODE(down$(1))<97
920 IF ok: ELSE PRINT down$&'is_'&across$
930 END IF
940 END FOR j
950 END FOR f: i$=INKEY$(#1,5000): STOP
960 ::

```


QL Today Volume 15 & Meetings

by Jochen Merz

Dear Reader,

Some months ago we were a bit undecided whether to continue with QL Today or not ... as we were not sure whether there will be enough readers and, also extremely important, enough writers!

The readership has shrunk a bit, as in the years before ... but as YOU are reading this issue, we would like to thank you explicitly for renewing. Maybe some renewals will arrive after I write this text - at least I hope so.

46 pages ... wow! This is much, much more than the planned 'average' of 32 pages!

As this issue will be sent from Austria, no major problem, as part of the extra pages will be made up by the cheaper postage again.

As the next issue will most likely be shipped from Austria too middle of December, please feel encouraged to send in material, articles, reviews etc. I am not complaining about too much material - as you can see, we are trying to do our best.

One article from Steve Poole will follow in the next issue, and a short article from George Gwilt... both really did not fit anymore.

And of course, the next part of Tony Tebby's view of the computing history.

But that's all we have - so we need MORE from you!

As you have probably read the issue completely by now, you may find that the questions written at the cover have not been completely answered yet.

And if you have had a look at the reverse side of this magazine - you'll find it is pretty empty ... at least not much contents there, deliberately!

Going back through the years, the reverse side of the magazine was filled with announcements of QL meetings all over the world.

Sometimes, I had problems getting all of them onto one page.

Eindhoven has been one of the most regular shows over the years, and our hope that these meetings would reappeared has not ended.

Vienna has been a nice meeting (although completely different to the meetings at Eindhoven or other QL meetings).

The report by Tony Firshman in this issue will give you an idea, and I have also written another completely different report (about the same weekend) for Quanta, which has been published already.

As some people feel it is probably too much to travel long distances nowadays "just" for the QL, the idea of combining it with sightseeing or a holiday makes a lot of sense.

I remember that many visitors of QL shows, especially in the USA, enjoyed the touristic plans around the QL meetings. I also remember the Berchtesgaden meetings well, where Friedeman Oertel arranged the shows.

Gerhard Plavec has done a great job in Vienna, setting up the website and hosting us visitors at his place.

It still leaves the question: where will be the next QL meeting, and when?

The Vienna meeting was created by talk at the Luzern meeting at the end of 2009 ... but so far there are no further plans for future meetings "out of the blue".

Which places would YOU like to see? It is probably difficult to top Vienna, but there are nice places all over the World.

Tony Firshman suggested a meeting in Paris.

My personal feelings about future shows:

I would like to see one or more meetings at Eindhoven again - and maybe after such a long pause, more visitors would show up too (we would have to find another after-show-Chinese as our standard one had closed before the last Eindhoven show).

I would like to visit Vienna again - maybe not next year, but how about 2012? I am sure many of the visitors who stayed only for a few days agree that Vienna deserves more time than a long weekend.

Somewhere in Switzerland again? Another part of Austria and/or Germany? The Salzburg / Berchtesgaden area is also very, very nice and could attract some QLers.

I would like to come to a QL show in England again - but as I explained already, I can't drive the long distance on my own anymore. I would like to combine it with other things in the UK too, not just drive over just for one day or two. Flying is not an option for me anymore.

I do remember nice meetings in Italy too... but it is a long drive...

Can we have your feedback please? How is your interest in meetings? Who is prepared to "do" something (find a location)? How do we reach QLers in the future? How do we find out how much interest exists in certain venues?

Many, many questions - we do need your replies!



The QL Show Agenda

